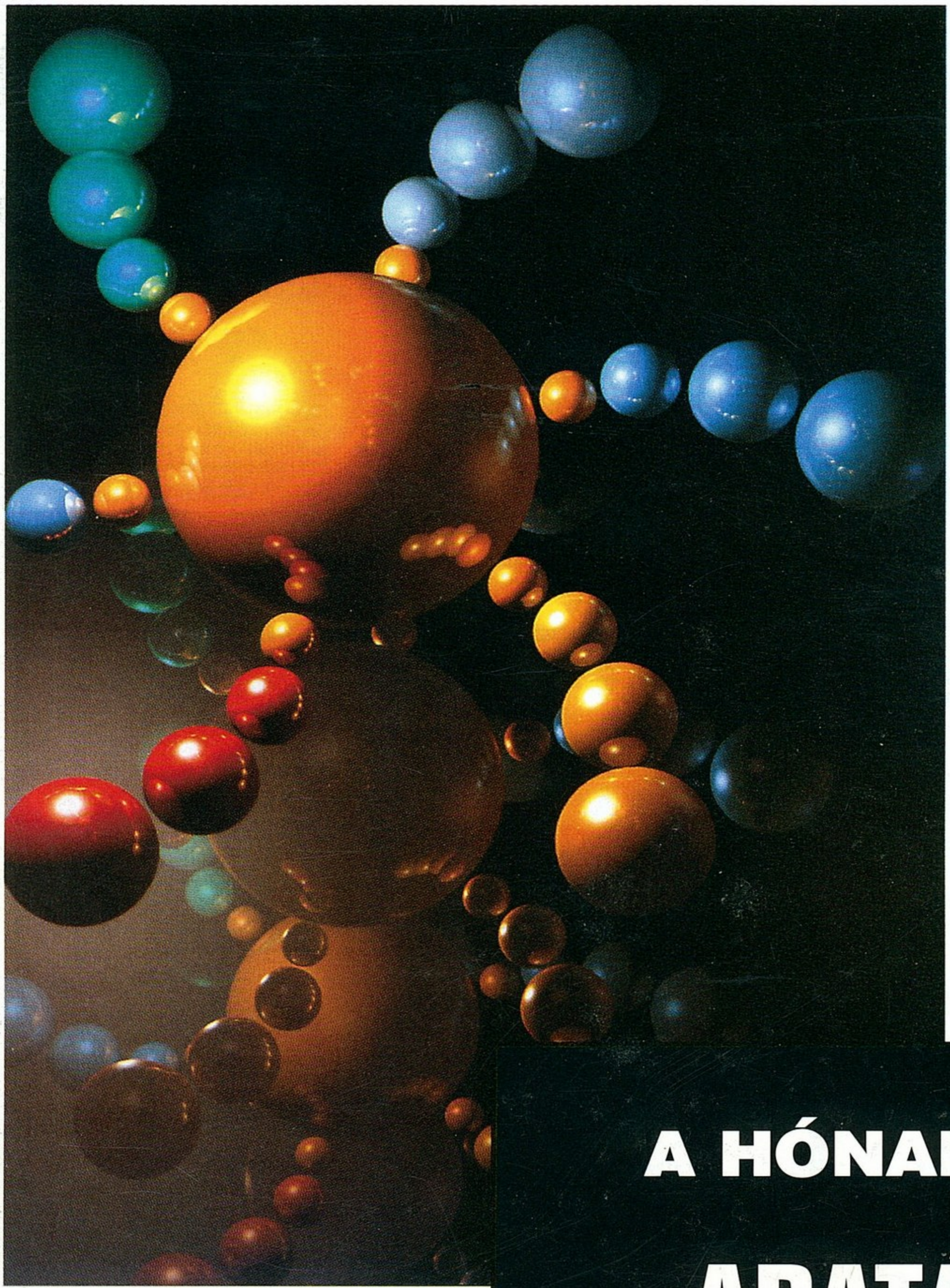
1996 / JÚLIUS

ÁRA: 356 FT

ÚJ ALAPLAP

SZÁMÍTÁSTECHNIKAI FOLYÓIRAT LEMEZMELLÉKLETTEL





Ikonkészítés képkivágással
Életjáték-programok
DOS-os installáló
Képkirakó
Amőba

CD-ROM diszkográfia

Az életjáték szenvedélye

A HÓNAP TÉMÁJA:

ADATÁRUHÁZ

A felügyelő

Beszél már a számítógépéhez?



CiscoPro Switching. A megfelelő sebesség!

Nem tudjuk fékezni magunkat.

Miközben arra törekedtünk, hogy a hálózatépítés technológiája a lehető legjobb legyen, nemcsak a router-ek piacán lettünk elsők, hanem, most már a Cisco a világ vezető switch gyártója is.

Ha jobban megismer minket, akkor ez nem lesz annyira meglepő. Bármi, ami a hálózatokat olyan gyorssá, megbízhatóvá és egyszerűen kezelhetővé teszi, mint a kapcsolók, biztos befutó nálunk. Mindig arra törekszünk, hogy a hálózaton keresztül történő üzleti kommunikációt még jobbá, még korszerűbbé tegyük.

A kapcsolók lehetővé teszik, hogy ez a törekvésünk végre munkához lásson!

Kiegészítve már meglévő rendszerét, a switching egy kézenfekvő továbblépési megoldást nyújt. Már működő hálózatát nem kell felszámolnia – pont ellenkezőleg! – ezzel még hatékonyabb lesz. És mindezek mellett (ami talán már túlzásnak is tűnhet), a CiscoPro család a skálázható megoldások irigylésreméltó választékát kínálja, hogy az Önéhez hasonló, növekvő szervezetek bővülő igényeinek megfeleljen.

Nem nehéz megkeresni azokat az embereket, akiknek az internetworking a szenvedélyük.

Nézzen utána a CiscoPro-nak az Interneten: http://www.cisco.com

Forduljon viszonteladójához:

WALTON NETWORKING LTD. — COMPUTER 2000

Albacomp ☎ (22)315-414 • Answare Kft. ☎ 467-1120 • Computronic ☎ 209-0538 • Digital Kft. ☎ (62)490-554 • Dr. Lan Kft. ☎ (46)341-838 • Linea 5 Kft. ☎ 251-8137 • Makrotrend ☎ 183-4356 • Műszertechnika Computer Rt. ☎ 260-4348 • Primus Net Kft. ☎ 153-4709 • Pro-Pid Kft. ☎ (30)561-312 • Unicomp Kft. ☎ 217-4170

Flag ☎ 274-2109 • Innet ☎ 371-0159 • Probit ☎ 269-7069 • Var ☎ 222-2827 • Networx ☎ 252-3444 • Elender ☎ 210-3044 • Probit ☎ 272-3453 • Rikvest ☎ 203-0533 • Rolitron Informatika ☎ 270-5120 • Professzionál ☎ 216-5300 • Delta Elektronik ☎ 180-0974

INFORMÁCIÓKÉRÉS: 0703 ▲

Novell®



GroupWise™

Ha versenyben akarnak maradni, *csapatban kell dolgozniuk a jó eredményekért. Ehhez azonban az elhatározáson és a megfelelő szakembereken kívül szükség van még valamire: olyan közvetítő közegre, amely lehetővé teszi* ***az összehangolt munkát.***

Ezt kínálja a GroupWise: e-mail, személyi határidőnaplók, csoportmunka-ütemező és Task-menedzser - egy alkalmazásban.

Egy csomagban, kedvező áron *beszerezhet mindent eszközt, amire a külvilággal való kapcsolattartáshoz szüksége lehet: GroupWise Client&Admin, Message Server (NLM), Async Gateway (NLM&OS/2), MHS Gateway (NLM), X/25 Gateway (OS/2).*

Az üzletet továbbra is Önnek kell vezetnie, de két dolgot a GroupWise-zal megtakaríthat:

Időt és pénzt.

WALTON NETWORKING LTD.

Walton Networking Kft.
1139 Budapest, Frangepán u. 8-10. Tel.: 344 38 38 Fax: 344 38 34
Walton Szegedi Iroda: 6723 Szeged, Sándor u. 1. Tel./Fax: (62) 490 424

INFORMÁCIÓKÉRÉS: 0732 ▲

ÚJ ALAPLAP

A Mikroszámítógép Magazin és az Alaplap hagyományait folytató számítástechnikai folyóirat

Megjelenik havonta, mágneslemez melléklettel

Főszerkesztő:
Faklen Pál

Főszerkesztő-helyettes:
Varga János

Szerkesztő:
Jakab Ágnes

Szerkesztőség és kiadó:
1539 Budapest I., Márvány u. 17.
Telefon: 156-3211 / 200, 214
Fax (manuális): 156-3211 / 201
E-mail: alaplap@mail.datanet.hu

Felelős kiadó:
Faklen Pál

Terjesztés:
Megyes Zsuzsanna

Hirdetésszervezés:
Árvai Katalin, Bogácsi Mária, Galyasi Hedvig, Pap Katalin

Külföldi hirdetések:
PubliCity
Reklám- és Médiaügynökség
1537 Budapest I., Márvány u. 17.
Telefon: 156-1182 Fax: 175-3539

A kiadó a hirdetések tartalmáért és a nyomdakészen kapott hirdetések formájáért (helyesírásáért) nem vállal felelősséget

Példányszámadatok hitelesítése:
Magyar Terjesztésellenőrző Szövetség
MATESZ
Ez a szám 10 300 példányban jelent meg

Nyomtatás:
Zalai Nyomda Rt, Zalaegerszeg
Felelős vezető:
Somogyi Tibor ügyvezető igazgató

Terjeszti:
A Magyar Posta Rt, a Nemzeti Hírlapkereskedelmi Rt, a Hírker Rt, a Kiadói Lapterjesztő Kft, számos számítástechnikai szaküzlet és más alternatív terjesztő

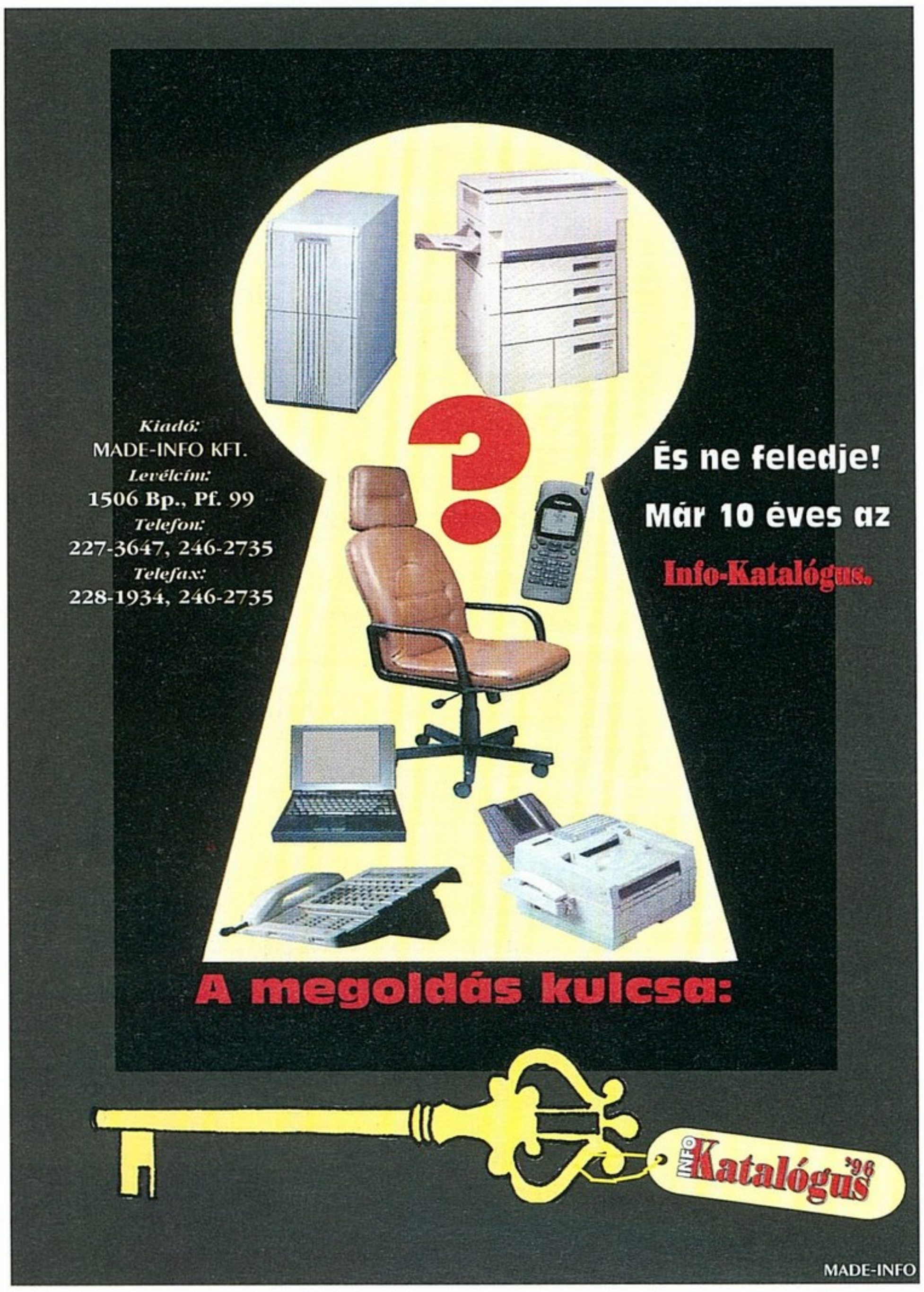
Előfizethető a kiadónál:
Új Alaplap Kiadói Kft,
1539 Budapest, Pf. 571
Bankszámlaszám:
OTP 11701004-20171649
Eladási ár: 356 Ft
Évi előfizetési díj: 3564 Ft

Külföldre terjeszti a Kultúra, H-1389 Budapest, Pf. 149

HU ISSN 1217-7598

A HÓNAP TÉMÁJA: ADATÁRUHÁZ

(Összeállította: Varga János)



INFORMÁCIÓKÉRÉS: 0715 ▶

INFORMÁCIÓKÉRÉS: 0722 ▶

INFORMÁCIÓKÉRÉS: 0701 ▶



INFORMÁCIÓKÉRÉS: 0728 ▶

Ég-e a ház?

A közelmúltban kisebb tűz támadt egy közintézményben. Azt annak rendje és módja szerint házilagosan eloltották. A jegyzőkönyvbe vélhetően ilyesmi került be: „Rövidzárlat következtében kigyulladt egy számítógép". Az eset nem túl gyakori, de mégis hétköznapi esemény. Érdekessé az teszi, hogy milyen számítógép is fogott tüzet... Nos, a hálózati adatforgalmat kiszolgálni hivatott szervernek lett elege a nyúzásból, és csendben elfüstölte a vezetékeit...

A konkrét tűzeset azonban csak ürügy arra, hogy átvitt értelemben tegyük fel a kérdés: ég-e a ház? Tervezhető-e, követhető-e, menedzselhető-e az adatforgalom például az elburjánzó Internet-használat közepette? Hány felhasználóig használható egy 100 useres Novell-hálózat? Mit kell tudnia egy adatbázisrendszernek ahhoz, hogy működése a hálózati eszközök optimális kihasználtságára is tekintettel legyen? Miként kommunikálnak egymással a hálózati forgalmat lebonyolító szoftvereszközök? Mit várnak el az adatbázisrendszerek a hálózati operációs rendszerektől — és fordítva: mit várnak a hálózati operációs rendszerek, menedzsment-szoftverek az adatbázisformátumoktól?

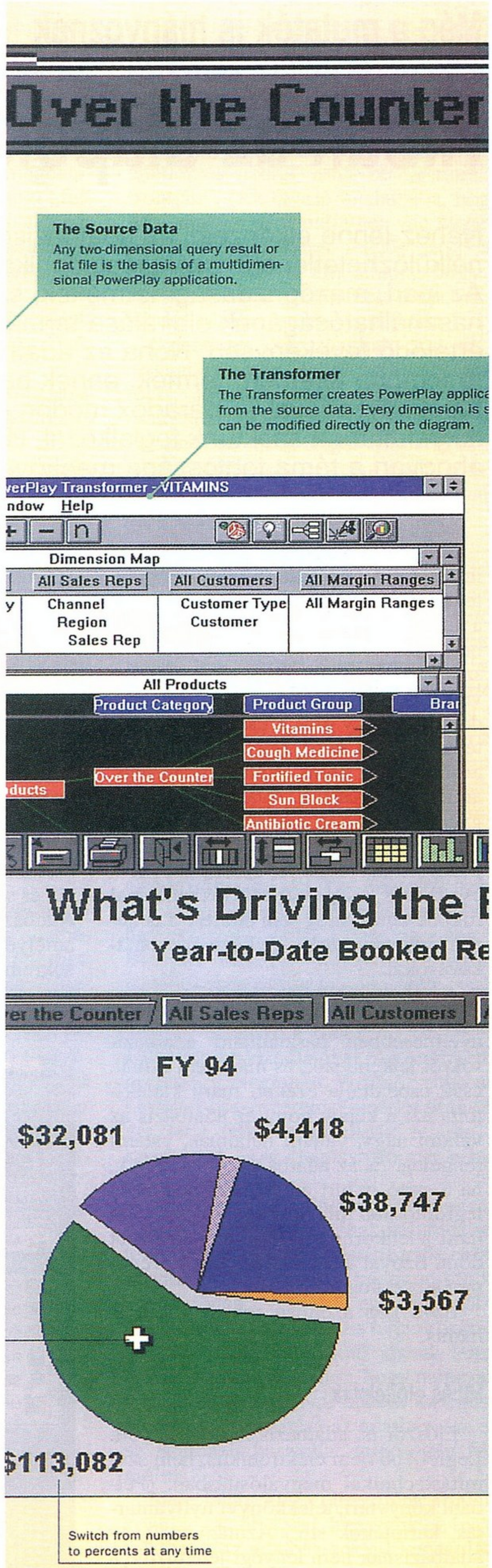
Arra természetesen senki ne számítson, hogy összeállításunkban a felmerülő össze kérdés megválaszolására sor kerülhet, de aki átrágja magát a mostani hónap témája cikkein, az ebben a témakörben igen tartalmas, nagyon friss és egyre inkább előtérbe kerülő információkhoz jut.

A számítástechnikának sarkalatos kérdése az adatbáziskezelés, amely akkor is fontos, amikor láthatatlanul, a háttérben zajlik. Sokszor azonban nagyon is a szemünk előtt vannak. Talán nem járunk messze az igazságtól, amikor az egész számítástechnikát egy mérhetetlenül nagy „adatáruháznak" tekintjük, amelyben adatmanipulációs osztályok egész sora hivatott segíteni munkánkat és mindennapi életünket — vagy éppen szórakoztatni bennünket.

Ha az adatbáziskezelés kerül szóba, a hozzáértő szakemberek egy része azt mondja: kár is vele foglalkozni, mert ha az adatbáziskezelés mint alapvető hardver/szoftver kérdés problémaként jelentkezik, az már régen rossz. Ezzel szemben a másik tábor szerint az adatbáziskezelés olyan kulcsfontosságú tevékenység, amelyet tökéletesen megoldani talán soha nem is lehet, mert a problémák jelentős része újratermelődik, legfeljebb másik — és nem mindig feltétlenül magasabb! — szinten.

Nem célunk sem az egyik, sem a másik tábor „bázisát" erősíteni, inkább egy körkép felvázolását kértük szerzőinktől: hol tart ma a nagy adatbázisok világa, s milyen fejlődési irányok rajzolódnak ki ezen a szakterületen.

Összeállításunk fókuszába olyan globális hálózati filozófiák megfogalmazásai és elemzései kerültek, amelyek piacvezető cégek kulisszái mögé nyújtanak bepillantást. És nekik azt is igazán tudniuk kell, hogyan érint mindez bennünket — mint adatbázis-alanyokat.



Még a mutatók is hiányoznak

Hibák az alapokban

Nehéz lenne elképzelni az adatbázisoknál fontosabb, nélkülözhetetlenebb számítástechnikai „terméket". Az ipari, mezőgazdasági termékek, szolgáltatások használhatóságának elbírálása természetes, magától értetődő tevékenység. Noha az adatbázis is (a legtöbb esetben) termék, ennek használhatóságával, hatékonyságával — paradox módon — csak felületesen, kutyafuttában szoktunk foglalkozni, távolról sem úgy, ahogyan a téma fontossága megkövetelné.

A nagy elektronikus adatbázisok az informatika korának valóban csodálatos produktumai. Ezek ma már nemcsak a tudományos haladás elősegítői, hanem a mindennapi élet nélkülözhetetlen szereplői is, ezért hálás téma minden velük kapcsolatos szenzáció. Nem csoda tehát, hogy erre a csemegére rávetette magát a sajtó. Így az adatbázis is arra a sorsra jutott, mint minden más, amit felkapnak: a róla alkotott kép gyakran eltorzul, eluralkodik benne a felületesség és az egyoldalúság, hiszen a dolgoknak csak az érdekes, csak a csodálatos, csak a szenzációs oldalával foglalkoznak, meg sem említve a nehezen érthető, szürke hétköznapi vonatkozásokat.

A sok sajtóhíradás, amely az egyre nagyobb, egyre gyorsabban, egyre kényelmesebben használható adatbázisokról lelkendezik, és mást sem csinál, csak csodáltatja ezeket, mára kialakította azt a képet, hogy az adatbázis az valami nagy, valami hatalmas, valami rengeteg. És az adatbázis ott kezdődik, ha a sajtó is hírt ad róla; az adatbázis legfontosabb tulajdonsága, hogy az valami jelenkori szenzáció, ami rekordot dönt. Szóval az adatbázis valami fenomenális dolog...

Ez a kép azonban többszörösen is hamis.

Hibás előítéletek

Először is, adatbázis mindig is volt. Legfeljebb nem elektronikus, nem számítástechnikai megvalósításban. (Például könyvtári, telekkönyvi nyilvántartás, kartotékok stb.) Azután pedig az adatbázisnak nem lényegi jellemzője a mérete, sem az, hogy hányan használják. Az adatbázisnak nem kell nagynak lennie, csak használhatónak.

Végül pedig az adatbázis nem rendkívüli, hanem mindennapi, természetes dolog, olyan, mint a lélegzetvétel: életünk tele van sok kisebb-nagyobb, szürke, egyszerű, de nélkülözhetetlen adatbázissal. Csak most, az elektronikus világban nem kezelik ezeket adatbázisként — e szemléletnek gyakran meg is issza a levét a társadalom és a tudomány.

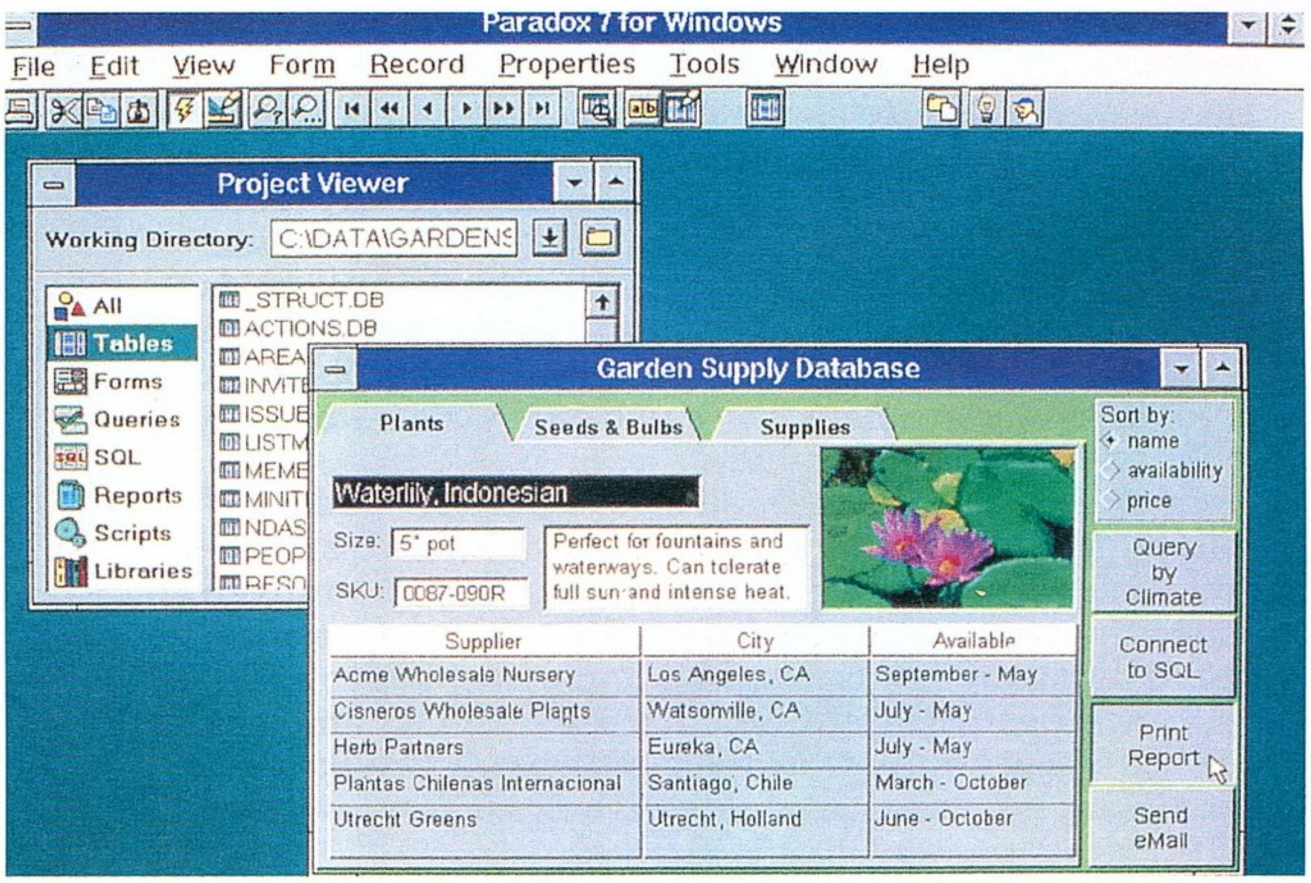
Oda jutottunk, hogy a bulvársajtó szellemi színvonala és értékrendje diktál, és sokszor már a szakma sem tekinti adatbázisnak azokat az adatbázisokat, amelyek nem elég óriásiak, nem elég sokfelhasználósak stb. Pedig adatbázis egy elektronikus gép bármely információtároló része, például a központi tár, benne minden, a számok és eljárások is, adatbázis a matematikai segédprocesszor ROM-ja, benne a már felfedezett és a még fel nem fedezett hibákkal, de adatbázis egy másodfokú egyenlet három együtthatója, a megoldóképlettel együtt. *Az adat az, ami adva van, az adatbázis pedig azoknak az adatoknak (számoknak, eljárásoknak stb.) az összessége, amelyek alapját, bázisát képezik egy vagy több feladat megoldásának.*

A szemlélet napjainkra meglehetősen eltorzult. Az adatbázis önálló, egyedi, elsődleges szerepbe került, függetlenítették mindentől, pedig csupán kiszolgáló része valaminek. Adatbázisa minden feladatnak van. És minden adatbázis létét feladatok indokolják.

A torzítás révén azután — feladatok nélkül — meg sem lehet fogalmazni azokat a kérdéseket, hogy az adatbázis milyen minőségű, azaz mennyire alkalmas azoknak a feladatoknak a megoldására, amelyek megoldását szolgálni lenne hivatott.

Mi van a tündöklő felszín alatt?

A félrebillent szemlélet helyrezökkentése érdekében nem lehet eléggé hansúlyozni: az adatbázis a feladatokért van, nem pedig azért, hogy csodáljuk nagyságát, elérési sebességét. Adatbázis önmagában nincs is, csak adatbázisa valamilyen feladatok megoldásának. Ne engedjük, hogy az adatbázis önmagában szerepelhessen, elkülönítve azoktól a feladatoktól, amelyek megoldásáért létrehozták, és amelyek nélkül lehetetlen az adatbázis minőségének, értékének megállapítása.



Mi az igazság az adatbázisok tótágast álló világában, ahol gyakran a kereskedelmi hírverés pótolja az alkalmassági vizsgálatot, és csak halovány remény van arra, hogy a közeli jövőben egzakt minőségmutatók alapján lehessen elbírálni, hogy melyik adatbázis mit is ér?

Ki kell már végre mondani, a tündöklő felszín alatt szürkeség, bizonytalanság van. „Az adatbázis" nem a szellem és a technika napsugaras csúcsa, mint ahogyan számosan igyekeznek a tudatlan tömeggel elhitetni, hanem nagyon is szürke, ködös, homályos, bizonytalan mocsár, ahol a mennyiségi mutatók sokszor demagóg módon érvényesülnek. (Csak a hazai példákból köteteket lehetne megtölteni ennek bizonyítására — folyószámlákról többszörösen levont összegek, az ingatlannyilvántartás és a cégnyilvántartás elképesztő anomáliái stb.)

Szemléleti alapok

A helyes szemlélet alapja tehát nem az adatbázis, hanem a feladat. A feladat megoldásához adottaknak kell lenniük követelményeknek, igényeknek, számoknak, módszereknek, anyagoknak, energiának, munkavégző operátoroknak stb. Ezeket valahonnan, legtöbbször raktárakból, készletező helyekről szerezzük be. Az adatbázis is egyfajta raktár, készletező hely.

Az eligazodáshoz nélkülözhetetlen, hogy kielégítő pontossággal jellemezni tudjuk a feladatot, a feladatmegoldást, a feladatmegoldási folyamatot, a feladatmegoldási lehetőségeket. Tudnunk kell válaszolni az olyan kérdésekre, mint például:

— Milyen a numerikus stb. adatellátottság?

— Milyen az adatmegbízhatóság, az adatminőség?

— Milyen a módszerekkel (ezek is adatok!) való ellátottság?

— Milyen a módszermegbízhatóság, a módszerminőség?

Ezek a kérdések (sokáig sorolhatóan) a feladat szempontjából vizsgálják az adatbázist, az adatokkal való alátámasztottságot, megalapozottságot — mint sikertényezőt, mint minőségmeghatározót (közkedvelt divatfordulattal „feltételrendszert"). Ha egy adott adatbázisból indulunk ki, akkor az adatbázis kiszolgálóképességét kell jellemeznünk.

— Milyen feladatokra, milyen jellemzőkkel (például hatékonysággal) alkalmas az adatbázis?

— Milyen feladatok megoldására, milyen jellemzőkkel tehető alkalmassá az adatbázis? Stb.

Mind ez ideig csak töredékekben léteznek olyan adatbázisok, amelyek a feladatokat (legalább a legfontosabbakat), azok adatellátási jellemzőit és lehetőségeit igyekeznének lajstromba venni. Adatbázis-nyilvántartó adatbázisokról is csak elvétve hallani: ezek feladata lenne részletesen jellemezni, hogy milyen feladatok megoldásánál melyik adatbázisrendszer milyen hatékonysággal használható.

Ami nagyon nincs...

Az adatbázis-használat minden munka természetes és nélkülözhetetlen alkotórésze, ennek ellenére nincs valóságos módszertana.

Bőszen folyik a verseny minél irdatlanabb numerikus és szöveges adatbázisok létrehozására. A képi és rajzi elemekből álló adatbázisok fejlesztésekor a mennyiségi szempontok háttérbe szorítják a kikerülhetetlenül fontos elméleti kérdések tisztázását. Ugyanakkor helyenként még az olyan egyszerű műveletek sincsenek kielégítően megoldva, mint például a keresés, az ellentmondás-kiszűrés vagy a hiányfelderítés.

Ezek sikere nagyrészt a modellek jó megválasztásától függ. Modellek pedig alig vannak. Az ellenőrzés-módszertan ismeretlen fogalom. A nagy pénzintézetek adatbáziskezelő programjait például független és erre szakosodott programellenőrzés nélkül vetik be „a munka frontján". (Az olvasó kitalálhatja, hogy hány magyar pénzintézetben van független, erre szakosodott felelős szoftverellenőrzés. Azt is kitalálhatja, hogy a bankok felügyeletére kötelezetteknek e kérdés eszébe jutott-e egyáltalán.)

Sztahanov már meghalt. De él a sztahanovizmus. Folyik az adatbelapátolás, minden adathordozóra, amire csak lehet, és várható, hogy az adatbázisoknál is, mint Hegel filozófiájában a mennyiségnek egyszer csak kedve támad „átcsapni minőségbe". (Vannak, akik ebben nem hisznek.)

A gépészek és a bádogos

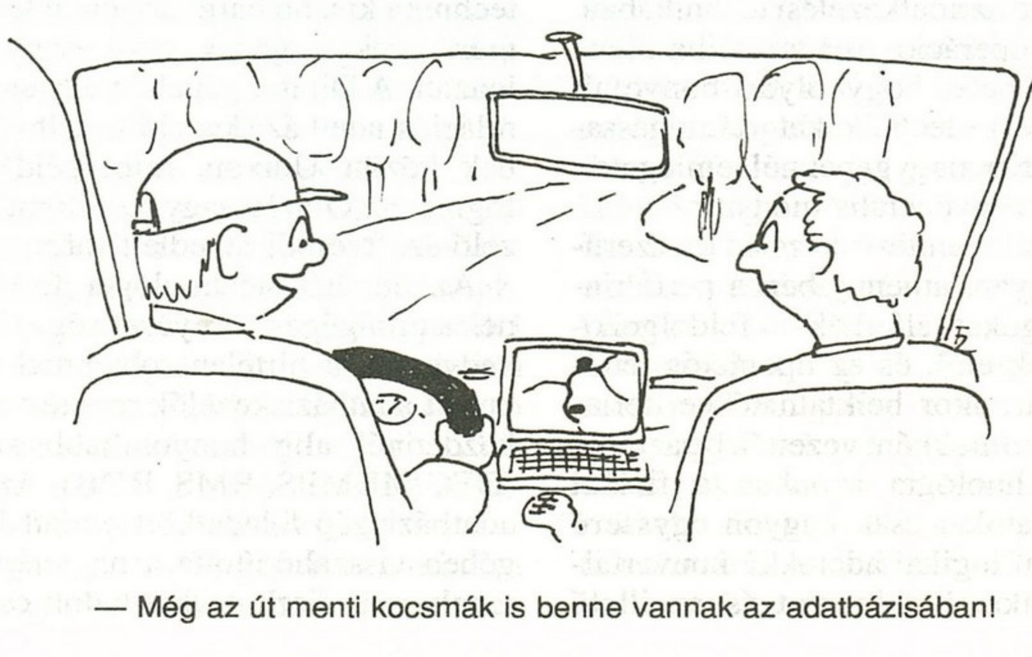
Engedtessék meg, hogy meglehetősen pesszimista gondolataimat egy megtörtént esettel fejezzem be. Egy nagy tudású gépészmérnök történetével.

Annál a vállalatnál, ahol dolgozott, a gépészrészleg főnöke egy bádogos volt. Egyszer szétszedtek egy sok fogaskereket tartalmazó finommechanikai szerkezetet (talán koppintás, illetve loppintás céljából), és az alkatrészeket egy kosárba rakták. Ott hevertek azok hetekig, mígnem elérkezett az újbóli összeszerelés ideje. Ekkor azonban a bádogos mérnök beosztottai pusztán tréfából betettek egy oda nem való fogaskereket is a kosárba, és figyelték, hogyan bajlódik a főnök a „kibővített adatbázis" használatával. Az eredmény mindenkit meglepett. A főnök több sikertelen összeszerelési kísérlet után elvesztette lelki egyensúlyát, feladta a küzdelmet, és az idegösszeroppanás határán lemondott.

Az adatokkal való megalapozottság a társadalom stabilitásának nélkülözhetetlen kelléke. Adatbázisaink nagy része azonban olyan, mint a bádogos főnök kosara. Másik részéből egy vagy több (fogas)kerék hiányzik. És teljes kosarak is hiányoznak. Nem kell jósnak lenni annak megállapításához, hogy nemcsak a magyar társadalom, hanem az egész emberiség stabilitása az adatbázisok és az adatbázis-használat minőségén fog múlni.

Ideje tehát, hogy elkezdjünk foglalkozni legalább azzal, hogy mik az igazi feladatok az adatbázisokkal kapcsolatban is.

Pogány Csaba



— Még az út menti kocsmák is benne vannak az adatbázisában!

Gépkategóriák és szoftverfejlődés

Adatbázis-erőművek

Az adatbázistechnikát korábban főleg szoftverproblémának tartották. Mára ez megváltozott. A hardver, az operációs rendszer, a hálózati technika és az adatbáziskezelő rendszerszoftver együttesen tudja csak megoldani az adatbázisokkal szemben ma felmerülő összes olyan igényt, mint például a napi 24 órán és évi 365 napon át megvalósuló folyamatos működés, évente összesen is legfeljebb perceket kitevő (gyakorlatilag nulla) leállási időt megengedő hibatűrőképesség, ami csak üzemelés közben történő berendezéscserékkel képzelhető el. Nem kevés feladat hárul a hardverre és a hálózatokra az adatok folyamatos mentése, illetve a hatékonyabb működést biztosító másolatállományok naprakészen tartása, vagy például az egyre rohamosabban szaporodó mobil alkalmazók kiszolgálása során.

Az adatbanktechnika, ahogy korábban a német nyelvterületről eredő megnevezést használták rá, a nagyszámítógépeken indult. Ezekben az időkben a 7,5 majd 29 Mbájtos kapacitású mágneslemezeken tárolt adatok már olyan hatalmas méretűnek számítottak, hogy kezelésükre külön szoftvertechnológiát tartottak szükségesnek. A korai időkben ez azonban vagy az invertált indexelési technika, vagy ennek spórolósabb megoldása, az indexelt szekvenciális adatelérési technika — lehetőleg operációs rendszeri — alkalmazása volt.

Később elkezdtek bonyolultabb kezelőalgoritmusokat is használni, amelyek már nem az operációs rendszer részét képezték, hanem a kifejezetten erre a célra készített adatbáziskezelő alrendszeri szoftverekét. Ilyen volt a láncolási technika (BOMP, D-BOMB), majd a relációs adatbázistechnika kialakulásával a B-fa (B-Tree) technika. Ekkorra a lemezegységek méretei már megközelítették a 100 Mbájtos méreteket is, ami a mai 2-3 Gbájtos PC-kben is közönségesen előforduló winchestertméretek mellett kicsit humorosnak tűnik (pedig a kapacitás növelését éppen az IBM sokat emlegetett winchesterfejtechnológiája tette lehetővé).

A nagygépeken a munka úgynevezett kötegelt, zárt géptermes rendszerben zajlott, ami az éles feldolgozásoknak rendkívül hatékony és biztonságos terepet nyújtott. A biztonság egyebek között éppen a géptermek zárhatóságából következett, nem úgy, mint ma, az Internet korában, amikor nem létezik zárt gépterem. A programok fejlesztése viszont rendkívül vontatottan haladt, a programozók munkájának hatékonysága az interaktivitás hiánya miatt elég alacsony volt.

Minigépek

A miniszámítógépek megjelenése gyökeres változást hozott a programozók életében: a programok fejlesztését online végezhették. Az interaktivitás terén elért kedvező változásért cserébe azzal kellett fizetni, hogy a minigépek erőforrásai ismét nyomorhelyzetet idéztek elő az adatkezelési technikában. E gépek operációs rendszereibe eleve nem fért bele, hogy olyan bonyolult adatkezelési technikákat támogassanak, mint a nagygépeknél emlegetett indexelt szekvenciális módszer.

Maga a berendezéskezelés korszerűsödött ugyan, amennyiben a perifériasajátosságokat elfedték a feldolgozóprogramok elől, és az operációs rendszerbe bármikor beiktatható perifériakezelő modulokként vezették be azokat. Ez a technológia azonban a fizikai szintű adatokat csak nagyon egyszerű szerkezetű logikai adatokká konvertálta. A fizikai blokkméret és az illető operációsrendszer-beli (félig hardvesnek is felfogható) logikai blokkméretek közti pufferelési technikát valósított meg.

Mire számíthatott tehát egy adatbáziskezelő rendszer írója a minigépeken? Egy szekvenciális be- és kivitelre, valamint egy állományon belüli adatpozicionálásra (logikai blokk, blokkon belüli relatív cím). Ebből kellett megoldani az adatok szekvenciálistól eltérő hozzáférésének beprogramozását. Az adatok felosztását olyan kisebb egységekre, mint a rekord, a hatékonyabb tárkihasználás érdekében pedig ezek összekapcsolását nagyobb egységekké (blokkokká), amelyek a fizikai tárolót a lehető legkevesebb hézagot hagyva töltik ki. A legszélsőségesebben takarékos megoldások minden létező bájttal spóroltak, úgyhogy az adatokat a fizikai tárolószerkezetétől függetlenül szorosan egymás mellé rakva tárolták. Különösen, ha az adatok logikai rekordmérete nem is volt fix méretű. A feldolgozás azután ugyancsak döcöghetett a hardverhatárokon szétvágódott rekordok eltérő elérhetőségi ideje miatt.

A nagyok folyományaként

A nagygépeken pedig ekkorra már hardverlehetőségekkel optimalizált gyors fizikai tárolási modelleket dolgoztak ki. Szét is választották az adatbázistechnikát három modellre: a koncepcionális, a logikai tárolási és a fizikai tárolási modellre. Az utóbbit a jobb operációs rendszerek, sőt a hardverek is támogatták.

Igaz, eleinte a nagygépeken indult el a relációs adatbáziskezelési technika is (IBM R és Sequel, majd Sequel-2 nyelv), de igazi sikert a minigépeken aratott, mert rendkívüli módon igényelte az interaktivitást, amelyet a nagygépes operációs rendszerek akkor még mindig elég „szőrmentén" támogattak. A siker mélyebb oka azonban a relációs technika kisebb hardverigénye lehetett, mert ezek a gépek ettől szenvedtek igazán. A Digital gépeken egy sereg jó relációs adatbáziskezelő indult el, többek között Unixon, mint például az Ingres, az Oracle vagy az állománykezelő szoftverből átvedlett Informix.

Az adatbázistechnológia fejlődésének a minigépes környezet végül is nem kedvezett, a hirtelen rohammal keletkezett adatbáziskezelők megrekedtek a kezdetinél alig bonyolultabb szinten (DEC MUMPS, RMS, RDB). Az igazi adatbázisgép feladatkört emiatt lényegében visszahódította a nagygépvilág, amely már diszkutcákat tudott csinálni

a 100 Mbájtos cserélhető lemezes tárolókból optimalizált elérést biztosító, magas szintű nyelvbe beágyazható alapszoftver-támogatással az adatbáziskezelős alkalmazói rendszerek fejlesztésére (Culprit stb.).

Mikrogépes korszak

A mikrogépek megjelenésével ismét javult az interaktivitás foka, mert a gépen lényegében senkivel sem kellett osztozkodni, azt ugyanis a szűkös erőforrások miatt csak egyetlen személy használhatta. Ettől ragadt rá a személyi számítógép elnevezés, hamarosan két kategóriában: otthoni hobbigépként és professzionális asztali gépként. A gyatra erőforrások miatt azonban megint kezdődött elölről a küzdelem. Már nemcsak az utolsó bájtot, de az utolsó bitet is meg kellett spórolni, ha kellően ki akarták használni e gépek lehetőségeit.

Ami az adatbázistechnikát jelenti, ebben a korban jelent meg az Ashton-Tate a Dbase-zel, amely bittérképeket használt, akár a Basic-alapú operációs rendszer a kis kapacitású (8, majd 5,25 collos) hajlékonylemezeken. A fejlődés szerencsére sokkal gyorsabb volt, mint a minigépek esetében.

Hamarosan megjelent az „ütésállóbb" winchester, amely a mai számítástechnika kulcsfontosságú szereplője. A PC-korszak vitte sikerre. Raádásul a fejlődés nem akadt el ugyanolyan módon, mint a minigépeknél. A PC-s winchesterekre írott adatbáziskezelési technológia hovatovább oda fejlődött, hogy az egész szakma fő mozgatórugójává vált.

Közben persze a PC-knek ehhez ki kellett nőniük önnön fogalmukat. Ma már nemcsak egyetlen személy kiszolgálására alkalmas erőforrásokkal rendelkeznek. Egy szabványos RISC mikroprocesszorokból felépített Cray, Silicon Graphics, Sun, Digital, Data General, Tandem, Motorola vagy IBM (sőt, Intel!) szuperparalel gépre ma már nem is mondjuk, hogy PC, pedig (néha sajnos) annak a technológiának a továbbfejlesztésén alapulnak, nem a nagygépeken, de nem is a minigépeken.

Sőt, a multi-mikroprocesszoros technológia kifejlesztésében sem mindenben a nagygépek nyomdokain haladtak. A különböző többprocesszoros, paralel gépi technológiákról azonban külön cikket lehetne írni, bár az Új Alaplapban már volt róla szó (de elég régen). A két legfontosabb multiprocesszoros megoldás a közös táron osztozkodó szimmetrikus paralel (SMP) gépek és a minden processzornak külön tárat adó masszív paralel gépek osztálya. Míg az előbbieknél elég egy speciális operációs rendszert használni, addig a másiknál a programokat teljesen újra kell tervezni a párhuzamos műveletvégzésből adódó lehetőségek komplex kihasználása érdekében. Az utóbbi gépekre ezért kevés a tényleg hozzájuk igazított szoftver. Az adatbázisgépekhez azonban mindkét modellt előszeretettel alkalmazzák, sőt, a Data General és a Sequent NUMA busztechnológiájának köszönhetően egy hibrid, mindkét megoldás előnyeit ötvöző gépkategória is születőben van.

Külön korszakot jelentettek a lokális hálózatok. A cégek — felismerve, hogy minden munkatárs asztalára kivihetetlen és értelmetlen lenne a drágább eszközök telepítése — a közös erőforrásokat speciális kiszolgáló gépre bízták, a többi gépet pedig „nagy" (5-16 Mbit/s) sebességű hálózattal hozzákötötték. Ez a korszak főként a Novell fényes pályafutásában nyilvánult meg (miközben az Asthon-Tate felszívódott, talán éppen ennek a fejlődési fokozatnak köszönhetően — a magányos farkasok ideje lejárt, bár nálunk még mindig sok kis Clipper-programozó utódja viszi a prímet, esetleg egy-egy régi ARCnetes NetWare tetején). A LAN elég hamar megnövelte az étvágyat a globális világhálózat kialakítására, amelyben az alkalmazók helytől függetlenül egyenrangú kiszolgálást várnak el.

Háromrétegű technológia

Az adatbáziskezelési technológia folyamatosan botlott bele a folyton kinőtt relatív címzési tartomány maximumába. Az IBM nagygépeknél először a 24-bites címtartománnyal volt baj. Most a 32-bites tartományt kezdik kinőni az ún. adatraktárak. A kiutat a 64-bites címtartomány (16 exabájt = 16 * 2^{60} bájt = 2^{64} bájt) jelenti, amelyet egyre több mikroprocesszor-család meghódít (Digital Alpha, MIPS RISC, Sun UltraSPARC és most a HP8000, majd később a HP—Intel közös ló).

Hol tartunk ma? A háromrétegű technológia (angolul a Three Tier Client-Server Technology) korszakában. Belátható, hogy a Novell NetWare LAN-ok nem hizlalhatók fel békából nőtt elefánt módjára. Még a Unixok sem (a Windows NT-k pedig a jelenlegi NT koncepcióval aligha). Ahogy mondani szokás: disztingválni kell.

A nagygépeket a downsizing jelszavával (a kisebb méretű elosztott eszközökkel való helyettesítéssel) kis híján felszámolták a „reklámszakemberek". Szerencsére nem ez történt. Manapság az IBM régi nagygépeket vásárol vissza, hogy azokat felújítva könnyebben tudja kielégíteni a nagygépek iránt újra feltámadó igényeket. Nos, a nagygépeknek valóban sok előnyük volt.

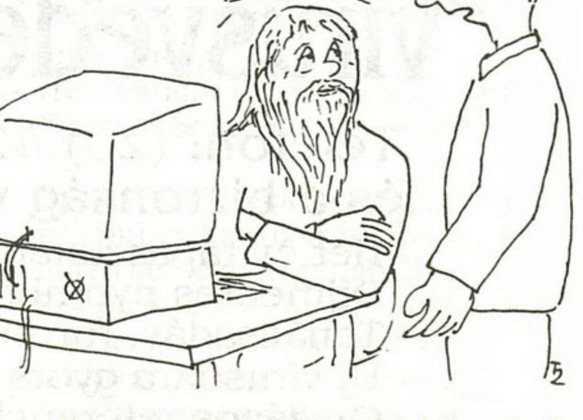
Az egyik, talán a legkevésbé elhanyagolható — és amire kevesen gondolnak —, éppen a biztonságtechnikát megkönnyítő zárt gépterem. Ennek egyik lecsapódása, hogy a leméretezett („downsizeingolt") gépeket elkezdték összehordani egy helyre, a régi klimatizált és zárható nagygépterembe. Mellettük pedig a munkaállomási és szuperszámítógépeket is behurcolták ezekbe a géptermekbe, alapvetően adatbázistechnikai feladatok támogatására.

A mostanra kialakult három szint tehát a következő:

1. A végfelhasználó terminálja, tipikusan egy asztali vagy mobil munkaállomás, amely esetleg hálózati számítógéppé (network computer) is elfajulhat, olykor otthoni, bonyolult multimédia-környezetben.

2. Adatállományokat kezelő, nyomtatókat vezérlő gépek, vagy egyéb kiszolgálógépek (szerverek), amelyekben általános, nem specializált adatbázistechnológiai feladatok is megoldhatók.

3. Adatbányák, amelyek igen nagy hardver-szoftver erőforrásokat igényelnek, és amelyek szolgáltatásait a végfelhasználók nem is közvetlenül, hanem a 2. szint szolgáltatásain át veszik igénybe. Elképzelhető, hogy ezek elemei a katasztrófatűrő sajátosságok biztosítására egymástól több száz kilométerre elhelyezkedő szuperszámítógépekből és speciális háttértároló alrendszerekből tevődnek össze (RAID — gyorsítás, tükrözés, üzemelés közbeni cserealrendszer, hierarchikus mentő-visszaállító alrendszer), amelyek tudják biztosítani a cikk elején emlegetett követelményeket. Ebben a hardver-, a szoftver- és a hálózati technológia egyaránt nélkülözhetetlen a megoldáshoz, mindegyiket optimalizálva, és távoli



— Maga belelát az én adatbázisomba?

működésfelügyeletre alkalmas megoldással ötvözve.

A fenti kategóriákból kilógó hardverek is vannak, mint például az Axis cégnek és másoknak közvetlenül a hálózatra csatolható, bárhonnan aktivizálható nyomtatószerverei, vagy a hasonlóan bárhonnan elérhető CD-ROM szerverek. Ezzel elkezdődött a teljesen osztott hálózati technológia kora.

Rendszertervezés és -szervezés

Azt még mindig kevesen akarják elfogadni, hogy az adatbázistechnika kulcsa nem a jó hardver — amelyből ma óriási a választék —, nem is a csoda-operációsrendszer (Unix—NetWare—Windows NT kakasviadal), nem is az évenkénti „kilopankráció" — az ezernyi újonnan beépített és persze hibás „képesség". Nem a mindenféle versennyel terhelt adatbáziskezelő szoftver a kritikus eleme az alkalmazói rendszerek sikerének, hanem a jó tervezés és elemzés. Azonban még a korábbi, főleg a nagygépes, centralizált adatbáziskezelős rendszerekhez kidolgozott SSADM és más hagyományos rendszertervezési technológiák grafikus interfészbe burkolt „korszerű" változatai sincsenek felkészítve, nemhogy a Three Tier (háromrétegű), de az egyszerű kliens-szerver technológia kezelésének a támogatására sem.

Az osztott feldolgozásnak az az egyszerű eleme, hogy az adatokhoz földrajzi koordináták is tartoznak, egyszerűen idegen azoknak a modelleknek a koncepciójától, amelyeket egy centrális adatbázis feltéteiezésével alkottak. A hálózati technika szempontjából azonban döntő fontosságú, hogy ahol az adat keletkezik, általában ott van vele a legtöbb munka, ezért földrajzilag a közelben kell tárolni, különben szükségtelenül leterheljük a hálózatokat. A távolabbi ponton elég volna a helyi adatokból származó kis volumenű, összegzett adat, éjjel pedig szépen át lehet pumpálni az archívumba az esetleg oda kívánkozó adatokat.

Hazai helyzet

Nos, állítólag a hagyományos adat–relációs (Data–Relation, D–R) technikákkal szemben a tárgy–szerep (Object–Role, O–R) technológiáké a jövő. Egyébként ilyen technika már azóta létezett Európában, mióta az amerikaiak az SSADM-et és társait ránk sütötték a világra. Nem először derül ki, hogy nem jó az irány...

Az utóbbi problémákra választ adó ilyen eszközök megjelentek a hazai piacon is. Az alkotók a fejlesztéskor már az emlegetett háromrétegű modellt vették figyelembe, és szigorú osztott adatbázistervezési munkát végeztek a programozás beindítása előtt.

A Watcom C++-ra és Gupta SQL Windowsra támaszkodó — és ilyen perspektívából tekintve talán a tavalyi Compfairen különösen méltán díjat nyert! — Infosys integrált vállalati információs rendszerről van szó. Az Infosyst fejlesztő magyar cég most felvette a kapcsolatot azzal az amerikai partnerrel is, amely az ilyen osztott környezetű vállalati újratervezési (reengineering) technológia kimagasló piacvezetője. A Sterling Software cég Business Processing Reengineering rendszerének óriási előnye, hogy a problémának a megoldását mindjárt modellezni is lehet, és egy-egy lefuttatás azonnal kiugrasztja a koncepcionális hibák nagy részét.

Zsadányi Pál

„Ha sikeres, utánozni fogják"

A követelmények „adatbázisa"

Vajon mi az oka annak, hogy az is adatbáziskezelő, amelyet számos kezdő programozó első zsengeként kiad a kezéből, és az a 3-4 csúcsteljesítmény is, amely kimagaslik a sűrű mezőnyből? Mitől „tuti üzlet" az egyik, és miért freeware a másik? Mi a közös és mi az eltérő e rendszerekben? Milyen szintjei vannak az adatbázis-hierarchiának?

Nem könnyű meghatározni, milyen minimális követelményeket kell kielégítenie egy szoftvernek, hogy kiérdemelje az adatbáziskezelő elnevezést. Talán több felhasználó szinkronizált, egyidejű kiszolgálása és a tranzakciókezelés az, amit például egy OLTP adatbáziskezelőnek tudnia kell. De vannak adatáruházhoz készült szoftverek, ahol inkább a többdimenziós adatkezelést helyezik előtérbe, és nem a tranzakciókezelést. A csúcson lévő adatbáziskezelőket (piaci részesedésük mellett) az is megkülönbözteti az alap-adatbáziskezelőktől, hogy egy több száz tulajdonságot tartalmazó listán a követelmények 90%-ának megfelelnek.

Emellett lehetnek egyedi megoldásaik, melyeket (ha beválnak) versenytársaik hamarosan szintén leutánozzák. Egy vezető adatbáziskezelőt tehát leginkább szolgáltatásainak széles köre emeli a többiek fölé. Megpróbáltuk csoportosítva felsorolni ezeket a ma általánosan elterjedt adatbáziskezelő-tulajdonságokat.

I. Adatok tárolása, adatbázis-üzemeltetés

1. Adattípusok
— Fix hosszú karakteres
— Változó hosszú karakteres
— Dátum
— Szám
— Bináris
— Tömbök
— BLO (binary large object) grafikák, dokumentumok tárolására
— Felhasználó által definiált adattípusok

2. Adattárolás
— Fájlrendszer
— Adattáblák átnyúlhatnak fájl- és diszkhatárokon
— Row device: az operációs rendszer fájlkezelő mechanizmusát kihagyó közvetlen diszkkezelés
— Csak olvasható állományok (CD-re, vagy hogy ne kelljen állandóan menteni)
— Data dictonary, az adatbázisban lévő objektumok, felhasználók, jogok, helyfoglalás, tárolt eljárások nyilvántartása
— Szinonimákkal új nevet lehet adni létező objektumoknak

3. Adatok írása
— Adatfájlt és naplófájlt is ír
— Az írás pufferelve halad, így egyes kiírások késleltetve vannak, de legkésőbb a commit utasítás kiadásakor legalább a naplófájlba kiírja
— Egy kiírásba több tranzakció adatai is bekerülhetnek (group commit)
— Tükrözött adat- és naplófájlok
— Naplófájlok automatikus archiválása diszkre vagy szalagra

4. Adatok olvasása
— Pufferelve történik, a gyorsító tár mérete tág határok közt állítható (néhány Kbájtos és néhány Gbájtos között), ezzel a fizikai I/O lényegesen csökkenthető

5. Mentés, visszaállítás
— Logikai mentéssel gépek között hordozható adatállományt kapunk
— Fizikai mentés (fájlmentés)
— Működés közbeni mentés (online backup)
— Rendszerösszeomlás esetén rendszerindításkor automatikus visszaállás
— Fizikai adatállomány elvesztése esetén visszaállás, a mentés és a naplófájlok segítségével
— Egyes állományok visszaállítása, miközben a rendszer többi részét használják
— Visszaállás múltbeli időpillanatbeli állapotra

6. Hordozhatóság
— Használható legyen különböző gépeken és operációs rendszereken (IBM VM, IBM MVS, UNIX, VMS, Windows NT, OS/2, Windows, NetWare)
— Kompatibilis, együttműködésre képes verziók működjenek az egyes operációs rendszereken
— Adatok és programok könnyen hordozhatók legyenek a különböző operációs rendszereken futó megvalósítások között

7. Párhuzamos hardverek kihasználása
— SMP (többprocesszoros gépek)
— MPP (masszív párhuzamos gépek), ahol több önálló gép van nagy sebességű vonalakon összekapcsolva, és szoftverszinten szinkronizálva (üzenetküldés, elosztott lock-kezelés)
— Közös diszkkel rendelkező önálló gépek (clusterek)

8. DBA eszközök
— Grafikus adminisztrációs környezet
— Felhasználói, elérési jogok szabályozása
— Helyfoglalás az adatoknak
— Potenciális és már fellépett problémák észlelése
— Rendszeres tevékenységek támogatása (automatikus mentés hetenként)
— Mindezt egy helyről, sok adatbázisra

II. Tranzakciókezelés, döntéstámogatás

1. Tranzakciókezelés
— Egymás utáni adatmódosító utasítások mind végrehajtódnak (commit), vagy egyik sem (rollback)
— Altranzakció (savepoint): vissza lehessen görgetni a tranzakció utolsó néhány utasítását (a definiált savepointig), megőrizve a tranzakció elején végzett változtatásokat, majd folytatni lehet a tranzakciót
— Tranzakciós monitorok támogatása

2. Egyidejű hozzáférés (concurrency control)
— Több felhasználó egy időben tudja írni, olvasni az adatbázist, de mindig konzisztens képet látnak az adatbázisról
— Mások változtatásait csak akkor lehet látni, ha azokat véglegesítették
— Ismételhető olvasás: egy tranzakción belül, ha ugyanazt a kérdést többször tesszük fel, ugyanazt az eredményt látjuk, akkor is, ha közben mások módosítani akarják az érintett adatokat
— Zárak (lock) kihelyezése: sor-, blokk-, táblaszinten írás-olvasásra

3. SQL-optimalizálás
— Statisztika-alapú: figyelembe veszi a táblák nagyságát, és hogy az egyes oszlopokban hányféle érték jelenik meg
— Hisztogram: az adatok eloszlását is figyelembe veszi (hány piros, zöld, kék stb.)
— Speciális SQL-ek optimalizálása, például csillagkeresés (star query), amelyben egy nagy táblához sok kisebb kódtábla kapcsolódik, a keresési feltételek nem a kódokra, hanem az értelmes fogalmakra vonatkoznak, de az eredményt a nagy táblából várjuk

— Figyelembe veszi a párhuzamosíthatóságot
— Befolyásolható (tanácsokkal, kész végrehajtási tervvel)

4. SQL-végrehajtás
— Kurzorokat kezel
— Módosítható az éppen lekért sor (update where current of cursor)
— Több sort kezel (beszúr, lekérdez) egyszerre, tömbösen
— Indexek (Btree, hash, bitmapped)
— Van csak indexeléréses végrehajtás (ha minden benne van az indexben, nem veszi elő a táblából a sort)
— Táblák összekapcsolására több módszere van (nested loop, sort-merge join, hash join)
— Együtt tárol táblákat, a gyakran kapcsolt mezők egyezése alapján (cluster)
— Ismeri az outer joint, amikor egyik vagy mindkét tábla összes elemét akarjuk látni, nemcsak azokat, amelyek a mezők egyenlősége alapján összekapcsolhatók
— A felhasználó definiálhat függvényeket, amelyekre SQL-ben lehet hivatkozni
— Módosítható, több táblán alapuló nézetek (updateable view of joins)
— From listában lehet egy teljes selectet megadni (inline view)
— Egy lekérdezés eredményének lekérése folytatódhat a tranzakció határán túl is (fetch accross commit)
— SQL-ek lefordított állapotban memóriában vagy az adatbázisban tárolhatók

5. Adatbetöltés
— Indexek karbantartása tömeges betöltés esetén is
— Létező táblára rá lehet még újabb sorokat tölteni
— Adatbázisblokk direkt írása (SQL-felület megkerülésével gyorsít)

6. Sorszámgenerálás
— Sorszám generálható, gyorsan, mások feltartása nélkül

7. Párhuzamos műveletek
— A rendelkezésre álló processzorok, diszkek kihasználása akkor is, ha csak egyetlen SQL fut a gépen
— Lekérdezés párhuzamosítása
— Adatbetöltés párhuzamosítása
— Indexek generálása

III. Alkalmazások támogatása, adatintegritás

1. Deklaratív integritás-szabályozás
— Nem kell programozni bizonyos ellenőrzéseket, elég deklarálni őket
— Tábla kulcsmezői (primary key)
— Egyik tábla hivatkozik egy másikra (foreign key)
— Egy sort ellenőrző feltétel (check constraint)

2. Triggerek
— Hozzáfér a régi és az új oszlopértékekhez
— Be tud állítani új oszlopértéket
— Letilthatók (nagy tömegű adatbetöltés, utólagos ellenőrzéssel), engedélyezhetők az SQL utasítás előtti és utáni triggerek; az SQL-en belül minden sor előtt és után lefutó triggerek, több azonos trigger
— Feltételes végrehajtás: a trigger csak akkor indul el, ha a megadott feltétel teljesül
— Triggerek kiválthatják egymást (cascade)
— Triggerek is a tranzakción belül futnak (commit, rollback a triggerek által végzett módosításra is hat)

3. Tárolt eljárások
— Tranzakción belül futnak még az elosztott adatbázisműveletek is (a „two-phase commit" védi)
— Más eljárásokat hívhat
— Csomagokba foglalhatók (package), amelyek elrejtik az eljárások törzsét, sőt a belső felhasználásra készült eljárásokat is
— Automatikus újrafordítás, a függőségi viszonyok alapján (ha X hívja Y-t, akkor X függ Y-tól, így Y megváltozása esetén X is újrafordítandó)
— Hivatkozhat távoli adatbázis adataira
— Hívhat olyan távoli eljárásokat (remote procedure call), amelyek ugyanazon a tranzakción belül futnak, mint a helyi hívások
— Paraméterezhető (IN, OUT, IN-OUT)
— Típusokat használó nyelv
— Vezérlésátadó utasítások: ciklusok, feltételes elágazás, eljáráshívás
— Overloading: azonos névvel, de eltérő paraméterekkel több eljárás is lehet
— Dinamikusan összeállított SQL-t lehet hívni belőle
— Lefordítva tárolódik az adatbázisban, és a memóriában marad futás után egy ideig (procedure cacheing)

4. Programozási felületek
— Előfordítók 3. generációs nyelvbe ágyazott SQL-hez
— Modulnyelv 3. generációs nyelvből hívható SQL-hez
— Közvetlen függvényhívási felület (API)

5. Nemzeti nyelvek támogatása
— Hibaüzenetek nemzeti nyelven
— 4GL programok, menük nemzeti nyelven
— Nemzeti konvenciók: dátumformátum, pénznem stb.
— Rendezés nemzeti karakterkészlet szerint
— Felhasználónként beállítható nyelvi környezet
— Karakterkészlet-konverzió kliens és szerver között (Unix szerveren terminálról is jól látszanak az ékezetes betűk, és ugyanaz az adat DOS-kliensen is jól jelenik meg)

6. Biztonság (security)
— Felhasználóazonosítás (név, jelszó)
— Elfogadja az op. rendszer felhasználóazonosítását, átveszi a nevet
— Objektumelérési jogok (írás, olvasás, végrehajtás)
— Rendszerjogok (mindent olvashat, bejelentkezhet, leállíthatja az adatbázist)
— A jogok szerepekhez, a szerepek felhasználókhoz rendelhetők

IV. Elosztott rendszerek

1. Hálózati kapcsolatok
— Kliens–szerver vagy szerver–szerver kapcsolat TCP/IP, DECNET, SPX/IPX, Named Pipes, LU 6.2, OSI4 ... protokollokon
— Protokollkonverzió (a kliens és a szerver eltérő hálózati protokollt használ)
— Hálózaton átmenő adatok titkosítása

2. Elosztott lekérdezések
— Egy lekérdezésben több adatbázisban (más gépen) lévő táblát is meg lehet nevezni
— Az optimalizálás figyelembe veszi a távoli adatokról tárolt statisztikákat is
— Az optimalizálás figyelembe veszi a táblák elhelyezkedését (ha mindkét tábla ugyanazon a távoli gépen van, akkor átküldi az egész lekérdezést, és csak az eredményt várja)

3. Elosztott adatbázisok módosítása
— Több adatbázisban egyidejű módosítás
— Kétfázisú commit védi a folyamatot (vagy minden változtatás végbemegy, vagy egyik sem)
— Transzparens a tranzakciókezelés (nem kíván külön programozást, csak a tranzakció végén commitot kell adni)
— Ha a hálózat hibája miatt egy véglegesítés elakad, a hálózat kijavítása után a tranzakció automatikusan befejeződik (vagy mindenhol commit, vagy mindenhol rollback)
— Hosszú ideig tartó hálózati hiba esetén lehetőség van kézi beavatkozásra, hogy a tranzakció valahogy lezáruljon (elengedje a zárolt objektumokat)

4. Replikáció
— Kiválasztott táblák másolása egy távoli adatbázisba
— Egyirányú, a másolatot ilyenkor csak olvasni érdemes
— Kétirányú, a módosításokat kölcsönösen átküldik egymásnak
— Szinkron vagy aszinkron (a módosítások egy sorba kerülnek be, és időnként továbbítódnak a célgép felé)
— A módosítások átküldése megtartja a tranzakciók és utasítások sorrendjét
— Tranzakciótartó átjárók (gateway-k)
— A kliens úgy látja az idegen adatbázist, mintha az számára ismert típusú lenne
— Feloldja az SQL-különbségeket

Mohay Tamás

Illustra-házasságból: Informix Universal Server

Pillantás a jövőbe

Az Informix jövőképéről ma már csak az Illustra felvásárlásának tükrében lehet beszélni. Az Illustra objektumrelációs adatbáziskezelő rendszer (ORDBMS) technológiája révén megnyíló lehetőségek ma még beláthatatlanok. 1996 végén megjelenik az Informix Universal Server, amely az Illustra legfőbb előnyét — tetszőleges komplexitású, a felhasználó által definiálható adattípusok kezelését — kombinálja az Informix nagy hatékonyságú, dinamikusan méretezhető adatbáziskezelőjével.

Az új szerver megjelenésével az Informixnak esélye van arra, hogy előnyre tegyen szert az összetett, dinamikus és interaktív (például web-lapok, kép, video, hang, térbeli 2D, 3D animációk, statisztikai modellek, pénzügyi és tudományos adatok időbeni viselkedésével kapcsolatos) adatok kezelése terén.

Közelmúlt

Persze a cég korábban sem szűkölködött az új ötletekban és azok megvalósításában. A dicsőségtáblán olyan képzeletbeli díjak találhatók, mint az első Unix-alapú RDBMS, az első OLTP szerver vagy az első multimédiás adatbázismotor.

Az 1990-es évek első fele az Informixnál igen jelentős technológiai fejlesztések jegyében telt. Háromévi műhelymunkát követően 1993 végén megjelent az OnLine Dynamic Server, amely az úgynevezett „dinamikusan méretezhető architektúrán” alapul. Az Informix teljesen újraírta adatbáziskezelőjét, az akkoriban még csak marginális szerepet játszó párhuzamos hardverarchitektúrákban rejlő teljesítmények kiaknázása érdekében.

A technológiai megújulás az alkalmazásfejlesztő eszközök terén sem maradt el. 1994 végén piacra került a NewEra, egy grafikus, objektumorientált, vizuális programozást és csoportmunkát támogató, második generációs kliens–szerver fejlesztőeszköz. Itt a második generáció egyik ismérve az alkalmazásparticionálás, amely lehetővé teszi az alkalmazás szétdarabolását a kliens és a szerver(ek) között. Ez a tulajdonság különösen nagyvállalati információs rendszereknél igen előnyös. Nagyvállalati igényeket elégít ki az is, hogy egyetlen osztálykönyvtár kicserélésével karakteres Unix terminálokon is futtatható a NewEra alkalmazás.

Az erőteljes fejlesztések során az Informix külön hangsúlyt helyezett régebbi szoftvereinek karbantartására, valamint a meglévő, működő alkalmazások modernebb környezetbe történő áthelyezésére.

Az Illustra-felvásárlás

A két céget néhány fontosabb vevő és hardvergyártó szinte rábeszélte a frigyre. Az Informix érdeklődését erősítette az a tény, hogy komplemens technológiákról lévén szó, az Illustra éppen abban jeleskedett, amiben az Informix nem, és viszont. Az Informix szervere mellett szólt a nagy teljesítmény, a méretezhetőség és a dinamikus kezelhetőség, míg az Illustra javára lehetett írni a tetszőleges adattípusok alkalmazásával elérhető korlátlan kiterjeszthetőséget.

Az Illustra ORDBMS hétévnyi kutatás és fejlesztés (Postgres projekt, Berkeley Egyetem) eredménye. 1995 végére már mintegy 400 ügyfélnél helyezték üzembe. Az Illustra egy adatbáziskiszolgálóból és tetszőleges számú, DataBlade nevű osztálykönyvtárból (modulból) áll. Az adatbázismotor a hagyományos RDBMS funkciók mellett az Illustra DataBlade technológiája révén lehetőséget nyújt új, tetszőlegesen komplex adattípusok létrehozására és intelligens kezelésére. Egy DataBlade modul adatstruktúra-gyűjteményt, a kapcsolódó adatmanipulációs (szükség esetén optimalizált) függvényeket, illetve opcionálisan indexelési módszereket tartalmaz.

Az Illustra jelenleg 10 DataBladehez tartozó több mint 100 adattípust és függvényt kínál. A DataBlade-ek a következő területeket támogatják: full text (szabad szöveg) keresés; 2D, 3D térbeli objektumok; időbeli folyamatok; statisztika; image (képfeldolgozás); video; web; vizuális információ-visszakeresés; gateway-k (Oracle, Sybase).

Objektum-relációs adatbáziskezelés (ORDBMS)

Az ORDBMS-ek részben relációs adatbáziskezelőknek is tekinthetők, mivel támogatják az SQL használatát, másrészt objektumorientáltak, a tetszőleges komplexitású adattípusok előállítása és kezelése szempontjából. Az ORDBMS-ek SQL felületének leírása, az SQL-3 még nem tekinthető szabványnak. Az SQL-3 az SQL-92 kiterjesztése felhasználó által definiált függvényekkel és operátorokkal. Ilyenek lehetnek az adott objektumok közötti hasonlósági, illetve távolságfüggvények.

Az Illustrában megvalósított objektum-relációs tulajdonságok a következők:

— Absztrakt adattípusok használata (struktúra és viselkedés specifikálása).

— Halmazok, tömbök, absztrakt típusok stb. tetszőleges mélységű egymásba ágyazása.

— Felhasználói függvények, operátorok definiálása adattípusokhoz (C vagy SQL szinten).

— Függvényekre is kiterjesztett költségalapú optimalizálás.

— Típusspecifikus indexelési módszerek.

— Polimorfizmus; egy- és többszintű öröklődés.

— Objektumok verziókezelése („időutazás”).

— Rule-ok, alertek használata.

Web-robbanás

A World Wide Web 1994 és 1995 folyamán elért drámai elterjedése a piaci szereplőket jelentős változtatásokra készteti. 1995 végén a nyugati világ cégei vagy üzemeltettek webszervert, vagy tervezési stádiumban voltak vele. A web a vállalaton belüli (intranet) és a külvilág felé történő (internet) információközlés standard eszközévé vált.

A web-kiszolgálók üzemeltetése azonban új problémákat is felvet. A web-lapok tartalmaznak képi információkat, és egyre gyakrabban video- és hangrészleteket is. A webbel kapcsolatos fejlesztések rendkívül gyors ütemben folynak. Új formátumok, tömörítési sémák jelennek meg. Kezdenek feltűnni a VRML standard nyelvet alkalmazó 3D web-szerverek is.

Ezt a nagy tömegű, dinamikusan változó, komplex adatot a hagyományosan működő RDBMS szerverek nem tudják kezelni. Az elmúlt 15 év során a kezelt adattípusok gyakorlatilag nem változtak. Ez erősen ellentétes a web világa felől jövő alapvető igénnyel, az új adattípusok mielőbbi támogatásával.

Világossá vált, hogy olyan adatbáziskezelőre van szükség, amely a komplex adatokat ugyanolyan hatékonyan képes kezelni, mint az egyszerűeket. Az objektum-relációs adatbáziskezelés (ORDBMS) pontosan ezt nyújtja: kombinálja az objektumorientált (OO) programozási és tervezési előnyöket egy rugalmas és a felhasználói igényekhez méretezhető (azaz hatékony) RDBMS-szel.

Még több komplexitás

A számítógépek teljesítményének növekedése, valamint a megnövekedett tárolókapacitás és hálózati áteresztőképesség (bandwidth) mind növeli a komplex adatok hatékony kezelésével kapcsolatos igények kielégíthetőségét.

A fekete-fehér képernyők kora lejárt. A mai felhasználók színekre, hangra, animációra és intuitív navigálási lehetőségekre vágynak. Azt is elvárják, hogy a számítógépes rendszer hatékonyan kezelje az üzlettel kapcsolatos összes adatukat (bármilyen összetettek is legyenek azok).

A megfelelő adatbáziskezelő hiánya egyes területeken már ma is veszteség forrása. Elég, ha az interaktív szórakoztatóipar (például CD-ROM-játékok) nehézségeire gondolunk. Ez a piac már 1994-ben nagyobb bevételt hozott, mint a hagyományos szórakoztatás, azaz a film- és videoipar együtt. Az interaktív játékok fejlesztői nagyságrendileg 10 és 100 gigabájt körüli adatot és százezres-milliós számú objektumot kezelnek. Megfelelő adatbáziskezelő nélkül ez a feladat szinte teljesíthetetlen az állandó és hosszadalmas keresések miatt.

Informix Universal Server

Az Informix és az Illustra technológiai összekapcsolódását megtestesítő Informix Universal Server az év végén jelenik meg. Az új szerver kombinálja majd a tetszőleges komplexitású adattípusok kezelését az Informix nagy hatékonyságú, dinamikusan méretezhető adatbáziskezelőjével.

Az Informix az adattípusok kiterjeszthetősége területén egy ideje maga is végzett kutatásokat, és elkezdte egy kiterjeszthető adattípusokat támogató, a DSA architektúrához illeszkedő API készítését is. Természetesen a fejlesztők jól ismerték és fel is használták a Postgres projekt eredményeit.

A két technológia összekapcsolása így sem kis feladat, a gyors integrálás főleg azért lehetséges, mert az Informix újraírta az adatbázismotort. Nem elhanyagolhatóak a két cég ügyfeleinek migrálásával kapcsolatos teendők sem. A lényeg az, hogy az ügyfelek minél gördülékenyebben álljanak át az új adatbáziskezelőre. A Universal Server ezért három fázisban jelenik meg az év folyamán.

Az év közepén megjelenő DataBlade fejlesztőkészlet segítségével olyan DataBlade-ek fejleszthetők, amelyek mind az Illustra jelenlegi kiszolgálójával, mind a leendő Universal Serverrel használhatók lesznek.

Szintén az év közepére várható egy Informix Gateway, amelynek révén az Informix-alkalmazások transzparens módon kezelhetik mind az Illustra szerveren tárolt komplex adatokat, mind a hagyományos, Informix adatbázisban tárolt adatokat. Év végére jön létre a két technológia teljes értékű egyesítése.

Jövő

Nagyon sok mai alkalmazási terület igényli az ORDBMS technológiát, például orvosi képfeldolgozó rendszerek, digitális könyvtárak, a film- és videoipar, az interaktív szórakoztató eszközök gyártói, a legtöbb tudományos célú adatbázis stb. Ezeken a területeken többnyire fájlkezelőket alkalmaztak, mert nem volt más.

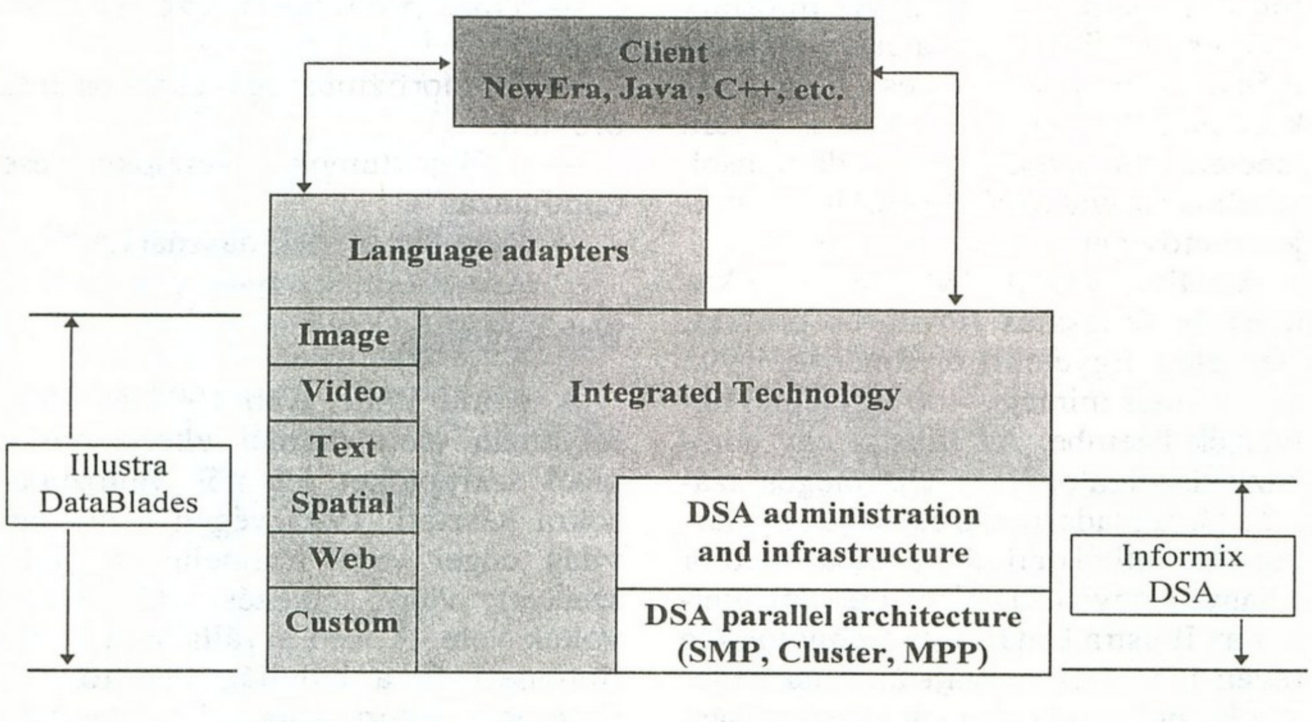
A mai, hagyományos RDBMS-alkalmazók nagy része is potenciális ORDBMS-felhasználónak számít. Tekintsünk egy tipikus biztosító társaságot, amely RDBMS-szel tartja karban az ügyfelek és a káresemények adatait. Az objektum-relációs technológia segítségével könnyen kiszűrhetnék a baleseti gócpontokat, a veszélyeztetett körzeteket, például a biztosítási díj „testre szabása" céljából.

A mai RDBMS-megoldások nagy része technológiailag amortizálódik. A hagyományos adatbázisok méretnövekedése a telítődési tendenciák miatt lelassul évi 10% körüli szintre. Ezzel szemben jelenleg ugyanannak a hardvernek a költsége (processzor, merevlemez, memória stb.) évente feleződik. Azaz egyre megfizethetőbbé válik az új technológia.

Ezek a tényezők mind az ORDBMS-piac megerősödését valószínűsítik. Az elemzők az 1996-ra jósolt 1 milliárd dolláros piac megtízszereződését várják 2000-re. Ekkorra várható, hogy az ORDBMS-bevételek meghaladják az RDBMS-piac bevételeit.

A WWW terjedése világított rá arra, hogy milyen intenzív változások következhetnek be egy sokáig lassan mozgó szakterületen. Senki sem tudja, mi jön legközelebb, de a gyors változásokban biztosak lehetünk. Mindenesetre a technológia már megvan.

Sándor Gábor



Az Informix Universal Server architektúrája

Az Oracle alkalmazkodó új technológiái

Algoritmusok minőségi ugrása

Az üzleti élet felgyorsulása széles körű adatelemzést kíván. Ez — szemben a hagyományos OLTP (Online Transaction Processing) alkalmazásokkal — elsősorban olvasás jellegű feladat (legfeljebb összegzett adatokat ír), így az egyidejűleg bejelentkező felhasználók szinkronizálása nem olyan nehéz feladat, s a pufferelt írás jelentősége is kisebb. Új igény azonban a nagy tömegű adat gyors feldolgozása. Erre részben új algoritmusok (aszinkron előre olvasás, adatparticionálás, bitmapped indexek, csillaglekérdezés, hash join, anti join, Gbájt méretű adateléréshez gyorsítómemória kezelése), részben az algoritmusok párhuzamosítása teszik képessé az Oracle-t.

Az aszinkron előre olvasás javítja a diszkműveletek és a CPU használatának átlapolódását, akkor is, ha a futásra kész processzek száma kicsi.

Az adatok particionálásával sok azonos szerkezetű fizikai táblát egy egységes logikai táblaként tudunk kezelni a lekérdezésekben, miközben az adatok betöltése, mentése, indexelése, purgálása a kezelhetőbb fizikai táblák szintjén történik (havi adatállományok). A lekérdezés-optimalizátor felismeri ezeket a logikai táblákat, akár táblánként eltérő elérési utat választ a view-ra megfogalmazott kérdés kiértékeléséhez. Fel tudja használni a táblákra tett megszorításokat is. Ha az X tábla dátum oszlopának értéke 1996-jan-01 és 1996-feb-01 közé esik, akkor abban nem keres 1995-re vonatkozó sorokat. A következő verziókban ez a lehetőség valódi táblaparticionálássá fejlődik majd.

Adatáruházak

A csillaglekérdezés gyakori művelet az adatáruházakban. Egy nagy táblában keresünk, amelyhez csillag alakban kicsi kódtáblák kapcsolódnak, és a keresési feltételek a kódtáblák oszlopaira vonatkoznak (nem a 3 színkódú, hanem a piros színű cikkeket keressük). Érdemes előbb az összes kódot kikeresni, és csak utána keresni a nagy táblában, egy konkatenált index segítségével.

A bitmapped indexeket olyan oszlopokon érdemes alkalmazni, amelyekben kevés érték fordul elő (termék színe, árkategória). Az ilyen oszlopokra megfogalmazott feltételekre hatékonyan lehet logikai műveleteket alkalmazni (termékszín = piros ÉS árkategória = III).

A hash join tekinthető a sort-merge join egy változatának, amikor nem rendezzük a két táblát, mielőtt a t1.o = t2.o feltételt kiértékelnénk, csak a mezőértékek hash osztályokba sorolásával válogatjuk kicsi, könnyen párosítható csoportokba a sorokat.

Az anti join a NOT IN feltétel gyors végrehajtási algoritmusa. A több Gbájt nagyságú gyorsító (cache) kiküszöböli a lassú diszkelérést, de ehhez hatékony tárolási struktúrára van szükség.

A párhuzamosan végrehajtható algoritmusok a sebesség növelésének egy másik lehetőségét nyújtják.

* Jól párhuzamosítható lekérdezési technikák: a teljes tábla végigolvasása, hash join, sort-merge join, egyes indexelt lekérdezések, eredmények rendezése.

* Párhuzamosan végezhető az adatbázis mentése, visszaállítása.

* Nagy adatmennyiségnél használható a párhuzamos betöltés, indexelés, eredménytáblák létrehozása (create table as select…).

* A párhuzamosított algoritmusok igénylik, hogy az adatok több diszkre legyenek szétosztva, és így egyszerre több adatforrást lehessen olvasni (ezek sokszor egy tábla részei).

* Az SMP gépeken az adatok szétosztásán kívül a hatékony processzek közötti kommunikációt is meg kell oldani, hogy a munkát tényleg szét lehessen osztani a rendelkezésre álló processzorok között.

* Az MPP (például: IBM SP2, N-Cube) gépeken további feladat a diszkek és processzorok távolságának megadása. Az Oracle gondoskodik róla, hogy az egyes diszkeken lévő táblapartíció adatait a lokális gépben lévő processzorok dolgozzák fel, és csak azután küldik át a lekérdezés koordinátorához a gépek közötti nagy sebességű hálózati kapcsolaton.

Sokféle adat kezelése

A hagyományos táblázatba foglalható adatokon kívül az Oracle sokféle adatot tud kezelni. Ezek az adatkezelők egyre szorosabban integrálódnak a hagyományos relációs adatbázishoz (SQL-lel lehet keresni bennük), és végül elvezetnek az objektumorientált adatbázishoz.

Oracle ConText Option. Szövegek tárolása, és bennük szavak keresése (szótő alapján) és a szókapcsolatok felderítése egyre fontosabb feladat, hiszen a mindent elöntő dokumentumtenger rendszerezésére, kulcsszavak szerinti besorolására egyre kevesebb erőnk van. A hatékony kereséshez előbb ki kell gyűjteni a fontos szavakat és előfordulási helyüket. A kibővített SQL-ben megfogalmazott keresés ennek felhasználásával találja meg a kívánt dokumentumokat. Feltétel lehet, hogy a megadott szavak egymáshoz közel (N szón belül) helyezkedjenek el a keresett anyagban. Angol szövegek gyorsolvasását egy nyelvi értelmező, információsűrítő modul segíti.

Térbeli adatok kezelésére fejlesztették ki a Spatial Data Optiont. Legszemléletesebb felhasználási területe a térképészeti rendszer, ahol gyakori feladat, hogy egy adott ponthoz közel eső objektumokat kell megkeresni. Hatékony megoldás, hogy a többdimenziós ponthoz egy HHCODE függvény egy mérőszámot rendel, amellyel könnyen meg lehet határozni a pontok távolságát. A módszer általánosítható, és más többdimenziós feladatoknál (tudományos számítások, idősorok) is alkalmazható.

Multimédia-alkalmazások számára hang- és képanyagokat képes tárolni és szállítani a Video Server Option.

Levelezés, időpontfeljegyzés, időpont-egyeztetés, dokumentumkezelés szervere a Messaging Option.

Web

Az Internet és a web-alkalmazások terjedésével természetes igényként fogalmazódott meg, hogy a statikus web-oldalak mellett dinamikus, adatbázisadatokat tartalmazó web-oldalak jelenjenek meg. Ezzel a kliens-szerver rendszerek egy új típusa jöhet létre. Mivel itt nincs állandó kapcsolat a kliens és a szerver között, a kliensnek a szerver szempontjából nincs előélete. Minden kapcsolatfelvétel egy önálló tranzakció vagy lekérdezés. Ami az előéletből fontos, azt a kliens az új web-kérésnél el kell, hogy küldje a szervernek.

Az Oracle Web Server három részre tagolható. Az első egy majdnem hagyományos web-figyelő (Web Listener), amely fogadja a bejövő igényeket, statikus fájlokat tud küldeni, és van CGI interfésze is. Az Oracle ki fogja bővíteni a támogatott web-figyelők körét más elterjedt HTTP rendszerű web-figyelőkre is.

A web-igények diszpécserprogramja (Web Request Broker Dispatcher) azonnal megkap minden bejövő igényt, és megkísérli a megfelelő kiszolgálóhoz továbbítani. Ha ez nem sikerül, akkor azt feltételezi, hogy ez egy hagyományos web-kérés, és visszaadja a web-figyelőnek.

Többféle végrehajtó program (Web Request Broker Executor Engine) várja a diszpécsertől a végrehajtandó feladatokat, a használt programozási nyelvnek megfelelően. Van PL/SQL, Java, LiveHTML, C++ végrehajtó, de a nyitott programozási felület lehetőséget ad újabb végrehajtók építésére is.

A végrehajtók állandóan futnak, nem kell állandóan bejelentkezniük az adatbázisba, ezzel jelentősen csökkentik a végrehajtási időt. A klienstől beérkező igények tulajdonképpen paraméterezett eljáráshívások, amelyeket az adott kiszolgáló végrehajt, majd az eredményt (egy frissen generált HTML-oldalt) a web-figyelőn keresztül visszaküldi a kliensnek (a Web Browsernek). A generált lapok általában az adatbázisból vett adatokat tartalmazzák, megfelelően formázva.

A választott architektúra miatt a web-figyelő nincs lekötve egy-egy hívással, ezáltal nagyobb áteresztőképességre („throughputra") képes. Ahogy egyre több alkalmazás települ az Internetre, egyre nagyobb szerepet kap a biztonság (security). Ennek megvalósítási lehetőségei: a browser azonosítása IP-cím vagy doménnév, felhasználónév/jelszó alapján. Az adatok titkosított átvitelét teszi lehetővé az SSL algoritmus (de ezt a browsernek is támogatnia kell).

Adatbázis-menedzsment

Ahogy az Oracle-felhasználók áttérnek a nagy központi szerverekről a kisebb gépekből álló, elosztott rendszerekre, a rendszerek karbantartása rendkívül nehéz feladattá válik. Az Oracle Enterprise Manager hivatott segíteni ezeken a problémákon. A DBA egy központi konzol előtt ülve monitorozza, diagnosztizálja az adatbázisokat. Jobokat indít el az egyes szervereken, amelyek adott időpontokban futnak le. Monitorozza a definiált események bekövetkezését, és megteszi a megfelelő lépéseket, hogy az eseményben jelzett gondokat elhárítsa. Lehetősége van a hálózati problémák elemzésére is.

A konzolról lehet monitorozó eszközöket indítani (statisztikák az egyes területekről, helyfoglalás a táblaterületen, lock monitor...), eseményeket definiálni, jobokat adott időpontokban indítani. Az esemény- és jobkezelést a szerverekre telepített intelligens ügynökök (Inteligent Agents) végzik, amelyek a konzolról kapják a parancsokat, és oda küldik a státusjelentéseket. Ez lehetővé teszi, hogy az ügynök elvégezze a munkát (lementse az adatbázist), amikor a hálózat hibája miatt a szervernek nincs kapcsolata a konzollal.

A transzparens gateway-technológia lehetővé teszi, hogy SQL-szinten Oracle-ből elérjünk RDB, DB2, DB2/400, EDA/SQL, APPC, Sybase, Ingres, Teradata, IBM DRDA, HP Turbo Image, Informix, RMS adatbázisokat. A Procedural Gateway Kit segítségével olyan gateway-t lehet írni, amely tárolt eljárásokat tud hívni.

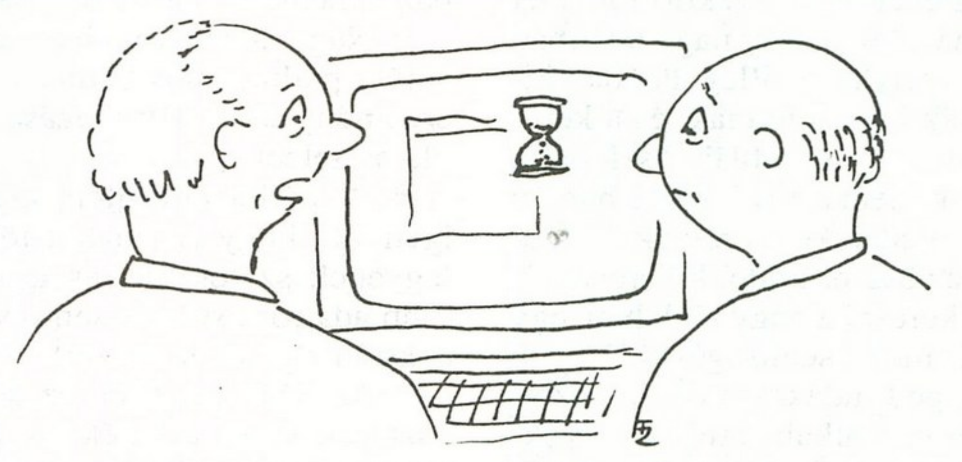
Biztonság

A biztonság vonatkozásában az Oracle kétféle jogot ismer. Az egyes táblákhoz, procedúrákhoz rendelt írási, olvasási, végrehajtási stb. jogokat (objektumjogok), valamint az általános bejelentkezés, bármilyen tábla olvasása, adatbázis leállítása jellegű jogokat (rendszerjogok). Egy adott felhasználó ezek kombinációját kaphatja meg. A könnyebb adminisztrálhatóság érdekében a jogokat szerepekhez lehet rendelni. A szerepekből újabb szerepeket lehet építeni, így alakulhat ki a jogok hierarchiája. Egy felhasználó több szerepet is kaphat, sőt mellette közvetlenül is kaphat jogokat. A jogok alkalmas kiosztásán túl be lehet kapcsolni a szelektív auditálást, amelynek révén követhető az egyes felhasználók tevékenysége, illetve észlelhetők a betörési kísérletek.

A felhasználó azonosítható jelszóval, vagy az azonosítást rá lehet bízni az operációs rendszerre is. A felhasználónak be kell jelentkeznie egy adatbázisba ahhoz, hogy adatokat érjen el. Nevezzük ezt hazai adatbázisnak. Ha másik adatbázisban tárolt adatot kér, akkor a hazai adatbázisban definiálni kell adatbázis-kapcsolatokat (linkeket) a távoli adatbázisokra. Egy link tartalmazza a távoli adatbázis elérési címét, egy ott érvényes felhasználónevet és jelszót. Amikor ebből a távoli adatbázisból adatot kér, akkor a hazai szerver bejelentkezik a távoli adatbázisba a linkben tárolt adatok segítségével, és elkéri onnan az adatokat, amelyeket továbbad a felhasználónak.

Nem elég biztosítani, hogy az adatokhoz ki férhet hozzá, mert az adatok a kliens és a szerver, illetve a szerverek között a hálózaton mennek át. Az Oracle Advanced Networking Option biztosítja a jelszavak kódolt átküldését, majd gondoskodik az adatok kódolásáról. Ezen túl az adatok lecserélés elleni védelméről is gondoskodik konzisztencia-ellenőrző kódolás alkalmazásával.

Mohay Tamás



— Nem kellett volna mégis inkább DOS-os adatbáziskezelőt venni?

Ismeretlen ismerős: IBM DB2/400

Megfejthető-e a fekete doboz „titka”?

Sokáig névtelenül használták. Az AS/400 adatbáziskezelője ugyanis integrált része az operációs rendszernek, szemben olyan versenytársak termékeivel, mint például az Oracle, a Sybase, az Informix vagy az Ingres. Nézzük, mi van a „fekete dobozban”.

A DB2/400 komponensei nem különíthetők el az operációs rendszertől, OS/400 nem létezik relációs adatbáziskezelés nélkül. Meglepő, hogy a felhasználók mennyire nem voltak tudatában annak, hogy mindennapi munkájukban egy relációs adatbáziskezelőt használnak. Felmérések szerint 40 százalékuk nem tudta, hogy birtokukban van az adatbáziskezelő, mégis természetesnek vették használatát. Lehet, hogy valamit csak akkor érzünk magunkénak, ha pénzbe kerül?

Jó, hogy nincs szükség adatbázis-adminisztrátorra, jó, hogy nem kell az adatbáziskezelőért külön fizetni, de rossz, hogy emiatt a felhasználók úgy érezték, kimaradnak valamiből. Ezért született meg a név: DB2/400.

Az adatbáziskezelő két arca

Az adatbáziskezelő egy váz, amely felel az adatdefiníciók karbantartásáért, az adatdefiníciók által előírt integritási szabályok betartatásáért, elvégzi az adatmanipulációs műveleteket, felel az adatbiztonságért stb. Mindezeket a műveleteket a DB/200 az OS/400 operációs rendszerrel integráltan végzi:

— A felhasználók az operációs rendszer által definiált felhasználók. Nincs külön nyilvántartás az adatbáziskezelőben és az operációs rendszerben.

— Az adatbázisok az operációs rendszer saját fájlrendszerében léteznek, az adatfájlok speciális adatterület-objektumokból (data-space object), az indexek pedig adatterület-indexobjektumokból állnak. Ezek az objektumok pedig az OS/400 MI (machine interface) szintjén érhetőek el. Az objektum szó nem csupán marketing-maszlag: az OS/400 objektumorientált operációs renszer. Pontosabban objektum-alapú (object based), az egyetlen lényeges különbség az öröklődésben van, az objektum-alapú rendszerek nem támogatják ezt a tulajdonságot. Az OS/400 objektumsablonokból (template) és objektumokból áll.

— Az adatbiztonságért az operációs rendszer objektumszintű védelme felel.

A sort még lehetne folytatni, de most inkább nézzük, miként felel meg a DB/2 a korábban vázolt követelmények legfontosabbjainak.

Táblák leírása és definiálása

Az AS/400-on az adatbázisok hagyományos definiálásának hagyományos módja az ún. DDS (Data Descriptions Specifications) fájlok készítése volt. Ma már nem vagyunk a DDS forrásfájlokhoz kötve, ha adatbázisokat akarunk létrehozni. Az SQL elterjedésével az AS/400-on is egyre többen használják az SQL adatdefiníciós nyelvét.

Egy adatbázis annyira relációs, mint amennyire annak tervezzük. Hiába dolgozunk relációs adatbáziskezelővel, ha tábláinkat rosszul tervezzük. ADB/200-ra két alapvető technika jellemző: az egyik a tábla sorainak egyediségét biztosítja (Primary Key), míg a másik a referenciális integritást (Foreign Key). Az utasításokat célszerű egy forrásfájlban leírni, és ezeket a Runsqlstm utasítással végrehajtani. Az utasítás részletes leírást ad minden parancs végrehajtásának sikerességéről.

A fizikai fájlokhoz hasonlóan lehet kétféle módon leírni a logikai fájlokat is. A logikai fájl az ismertebb SQL szabványban view-nak nevezett objektum, mely a fizikai fájlok által leírt adatok különböző nézeteinek megvalósítására való. Nézetekkel fizikai fájlokat kapcsolhatunk össze, mezőkön típuskonverziót hajthatunk végre, kiválasztási kritériumokat szabhatunk meg, vagy éppen mezőszintű adatvédelmet biztosíthatunk.

Adatvédelem

Az AS/400-as védelmi rendszere alapvetően az objektumok (itt fájlok) szintjén határozható meg.

A védelem kettébontható:

1. Objektummenedzsment (attribútumok lekérdezése, objektum átnevezése, mozgatása könyvtárak között, törlés, trigger vagy referenciális integritás definiálása).

2. Objektumok tartalmának manipulációjára vonatkozó engedélyek (írás, olvasás, módosítás, törlés, végrehajtás).

A logikai fájlok további biztonságot nyújthatnak rekord-, illetve mezőszintű védelemmel.

Adatbiztonság

A naplózáshoz, amely a commitment controll alapja, nincs szükség az adatbáziskezelő által definiált fájlokra — mint amilyen például az Oracle redo log fájlja — vagy kontrollfájlra, amely ezeket nyilvántartja.

Az AS/400-on a naplózást két rendszerobjektum teszi lehetővé:

— Journal (naplóleíró): a naplóobjektum tartja nyilván a naplózott objektumokat.

— Journal receiver (naplófogadó): tárolja a naplózott változtatásokat.

Fontos tudni, hogy naplózni nemcsak a fizikai fájlokat lehet, hanem az indexterületeket is. Egy abnormális leállás után az OS/400 az érintett indexeket automatikusan újraépíti, ami nagy adatállományoknál sok időt vehet igénybe. Ezen segíthet az indexek naplózása, miáltal az indexek töredék idő alatt újraépülnek.

Triggerek, tárolt procedúrák

A trigger egy speciális program, amelyet általában user-exitként szokás nevezni, meghívásáról az adatbáziskezelő saját maga gondoskodik. Az aktiváló esemény egy rekord beírása, módosítása vagy törlése lehet, ezenkívül meghatározható, hogy az akció elvégzése előtt vagy utána fusson trigger programunk. Például egy megrendelés-nyilvántartó program a megrendelés rögzítésekor automatikusan aktivál egy trigger prog-

ramot, amely ellenőrzi a raktárkészletet, és amennyiben szükséges, az új rendeléshez a szükséges alkatrészek beszerzését kezdeményezi a szállítóktól.

Az így definiált trigger programok a tárolt procedúrákkal kliens-szerver megoldásaink alapját képezhetik, velük írhatjuk le üzleti szabályainkat, amelyek e szabályok hatékony betartását biztosítják. Hatékonyak, mivel az AS/400-on mind a trigger, mind pedig a tárolt procedúrák natív AS/400-as programok, nem pedig egy, az adatbáziskezelő által interpretált nyelv szűkös utasításkészletével írt kóddarabkák (mint amilyen például a PL*SQL az Oracle DBMS-nél).

Párhuzamos adatbázisfunkciók

A párhuzamos struktúráknak három műfaja létezik:

1. Osztott memória (alias osztozzunk mindenen), ebben a megoldásban a processzorok osztoznak egy közös globális memórián, amelyen keresztül kommunikálnak a külön processzorokon futó adatbáziskezelők. A megoldás korlátját nem a processzorok sebessége adja, hanem a közös memória elérésének sebessége. Ez minden hozzáadott processzorral jelentősen lassul, így tehát nem jól skálázható.

2. Osztott lemezek (alias osztozzunk az adatokon). Itt minden processzor saját memóriaterületén dolgozik, és a hálózati funkciók gondoskodnak a lemezterület eléréséről. Skálázhatóság szempontjából a nehézséget az újabb lemezek hozzáadásával együtt járó komplexitás növekedése, a közös lockolás, illetve a commitment controll problémái jelentik.

3. A harmadik megoldásban, amelyet az AS/400 fejlesztői is választottak, az egyes processzorok nem osztozkodnak sem a memórián, sem a lemezeken, ehelyett az SQL-hívásokat szétosztják logikai alutasításokra, és ezek végrehajtását egymástól függetlenül végzik a hálózatba kapcsolt node-ok.

A párhuzamos feldolgozás alapegységei AS/400-as node-ok. A node-okat, amelyek hálózatba (bármilyen protokollal) kapcsolt AS/400-as rendszerek, csoportokba rendezhetjük. Minden egyes node a csoportban egy partíciós számot kap, 0-tól 1023-ig. Ebből a jobbak már ki is találhatták a folytatást: hashing algoritmus (mikrokódként implementálva). Az algoritmus inputként a particionált tábla egy kulcsát kapja, ebből képezi a 0 és 1023 közötti számot, amely meghatározza, melyik node érintett az adott tranzakcióban, és annak irányában továbbítja az adatbázisszerver az SQL-kérést. A csoportba terelt node-ok között alapértelmezésben egyenletesen oszlik meg a terhelés, de amennyiben akarjuk (és miért ne akarnánk, feltéve, hogy az egyik masinánk multiprocesszoros AS/400-as), megbonthatjuk az egyensúlyt, és a rekordok felét erre a gépre tereljük.

A létrehozáskor minden node-hoz egy kétbájtos szám rendelődik. Ezt a számot használhatjuk fel egy partíciós fájl létrehozásához, amely egyetlen kétbájtos, decimális értékeket tartalmazó mezőből áll. A fájlt feltöltjük 512 rekorddal, ez a multiprocesszoros rendszerünk node-számát tartalmazza, a maradék 512 rekordot egyenletesen felosztjuk a két másik node között, majd ismét kiadjuk a Crtnodgrp parancsot, most azonban megadva a partíciós fájlt is. Ezzel a megoldással azonban nem vezérelhetjük, hogy mely rekordok melyik node-ra kerüljenek.

Referenciális integritás DB2/400-on

Az operációs rendszer V3R1 verziójától kezdve része az adatbázisnak. A korábbi verziók alatt az adatbázis integritásának megőrzése az alkalmazások felelőssége volt, ez ugyanakkor azt is jelentette, hogy az adatbázis szerkezetének egy jelentős része rejtve maradt a felhasználói programokban. Az új adatbáziskezelővel a fájlok közötti kapcsolatok leírása átkerül a programokból az adatfájlokba — ezáltal rövidebbé és megbízhatóbbá válik a fejlesztés és a karbantartás. A DB2 az RI-t fizikai fájlmegszorítások (constraintek) formájában valósítja meg.

Három típust különböztetünk meg, ezek közül az első kettő nem referenciális (de létezése szükséges az RI-hez), a harmadik referenciális.

1. Elsődleges kulcs megszorítás (primary key constraint).

2. Egyedi kulcs megszorítás (unique key constraint).

3. Referenciális megszorítás (referential constraint).

Az első két megszorítás egyetlen fájlra vonatkozik. Az adatbáziskezelő ezt egy egyedi kulcsra épített index (logikai fájl) formájában valósítja meg, amely lehet már létező (például korábban DDS-szel létrehozott logikai fájl). Amennyiben ilyen nincs, a DB2 létrehozza. Egy fizikai fájlhoz legfeljebb egy elsődleges kulcs tartozhat, amelyhez további egyedi kulcsokat definiálhatunk.

Mi ebben az új? A korábbi gyakorlattal ellentétben, amikor a fájlszintű adatintegritást logikai fájlokon keresztül érvényesítettük, a változás az, hogy ezek a megszorítások most egy helyen, a fizikai fájllal tárolódnak. Sokaknak ismerős gyakorlat a következő: egy újonnan megírt program elszáll egy rekord írásánál. Ekkor derül csak ki, hogy a fizikai fájlra épített n logikai fájl közül az egyik tartalmaz egy egyedi kulcsot, amelyet a program tervezésénél nem vettünk észre. A merészebbek persze ilyenkor hanyag eleganciával kitörlik a logikai fájl DDS-éből a Unique kulcsszót, és újrafordítják, az óvatosabbak megpróbálják felvenni a kapcsolatot a logikai fájl írójával, és kideríteni, vajon csak pillanatnyi felindulásból elkövetett terméke-e az egyedi kulcs, vagy lényeges része az adatbázis struktúrájának.

A contstraintek használata ezzel szemben egy helyen, a fizikai fájl objektummal tárolja ezt az információt, leegyszerűsítve az adatbázis tervezését és karbantartását.

A referenciális integritás akkor kezdődik, amikor már több fájl közötti kapcsolatot akarunk leírni.

Kliens-szerver alkalmazások támogatása

Adatbázis-triggerekkel, illetve tárolt procedúrákkal az egyenlet szerveroldalát már érintettük. Alapvetően a következő kliens-szerver technológiák érhetők el az AS/400-on:

— Remote SQL (távoli SQL) API. Lehetővé teszi klienprogramok (OS/2, Windows) számára AS/400-as adatbázisok lekérdezését, programozása nehezebb, mint az ODBC API-é, és teljesítménye is rosszabb. További hátránya, hogy nem vált iparági szabvánnyá, s az így fejlesztett alkalmazások nem lesznek hordozhatók.

— ODBC API. Teljes 2. szintű támogatást kapunk, ha CA/400 környezettel dolgozunk. (A CA/400 az IBM kliens-szerver hálózati szoftvercsomagja.) AS/400-on tárolt procedúráink a kliensalkalmazások számára nem csupán Update, Insert, Select utasítások sikerességét tudják visszajelezni vagy egy-egy paraméterét visszaadni, hanem egész rekordkészleteket adhatunk vissza a hívó Windows klienprogramoknak.

— DPC (Distributed Program Call) API. Szintén AS/400-on lévő programjaink meghívására alkalmas, mindezt nagyon jó hatékonysággal végzi. Szerverprogramjaink beépített SQL-utasításokkal javíthatják az alkalmazás teljesítményét.

Molnár Balázs

Borland Database Engine

Egységes programozói interfész

Az alábbi cikk azokat az okokat elemzi — a funkcióleírásra koncentrálva —, amelyek révén a BDE napjaink talán a legelterjedtebb windowsos adatbázis-motorjává nőtte ki magát.

A fejlesztők előtt jól ismert az a kihívás, amelyet a jövőbeli fejlesztésekre gondolva egy feladat megoldásának általánosítása, kiterjesztése jelent. A Borland is hasonlóval állt szemben, amikor elhatározta termékei közös adatbázis-motorjának kifejlesztését. Nagyjából a következő elvárásokat fogalmazták meg:

1. Talán a legfontosabb egy egységes programozói interfész (IDAPI) kialakítása volt, amely független a BDE alatt lévő tényleges adatforrástól. Azaz a fejlesztőnek ne kelljen állandóan kézikönyveket bújva szem előtt tartania az éppen általa használt adatfájlformátum specifikumait. (Például: ugyanazzal az IDAPI függvényhívással lehessen egy Dbase-, egy Paradox-állomány vagy akár egy Oracle SQL-tábla utolsó rekordjára pozicionálni.) Természetesen ezt az általánosítást sem lehetett a végletekig vinni, hiszen például a Dbase-állományok nem alkalmasak OLE-objektumok tárolására, így nem is várhatunk el a Paradoxhoz hasonló IDAPI függvényeket, amelyekben van ilyen mezőtípus. A BDE-vel való ismerkedésünk során ezért olyan gyakori hibaüzenet a „Capability not supported"!

2. Egyszerre több windowsos alkalmazás is használhassa ugyanazokat a BDE erőforrásokat. Legyen egyszerűen konfigurálható (nyelvi meghajtók, Windows-beállításoktól független dátum-, idő- és pénzformátumok), könnyen hordozható, egyszerűen telepíthető. De ne legyen a háttérben egy futtató (lassító) környezet.

3. Biztosítson adathozzáférést a lehető legszélesebb skálán. Akár ODBC drivereken keresztül, akár az illető adatbáziskezelő honi, natív SQL nyelvén (SQL-link). Tegye lehetővé a kliens/szerver rendszerek fejlesztését ugyanúgy, mint asztali vagy LAN környezetben futó alkalmazások készítését. Tranzakció-kezelése jól illeszkedjen az SQL-szerverek izolációs szintjeihez. Legyen világos és konzisztens a hálózatkezelési filozófiája. Legyenek az IDAPI segítségével keszített alkalmazások felfelé és lefelé is egyszerűen méretezhetőek (upsizing, downsizing).

4. A kidolgozásnál szem előtt kell tartani a BDE későbbi módosíthatóságának, fejleszthetőségének, esetleg új operációs rendszer alá történő átírásának lehetőségét. Ezt az igényt úgy is megfogalmazhatjuk, hogy a BDE legyen messzemenően OOP-szemlélettel tervezett és kivitelezett nyitott termék, amely integrálja a Borland e téren szerzett tapasztalatait. Gondoljunk csak a Paradox-Engine-re, mint talán legfontosabb előzményre, illetve arra, hogy amikor a fejlesztés elindult, a Borland szlogenje még „The Leader of OOP" volt a mai „Upsizing Company" helyett.

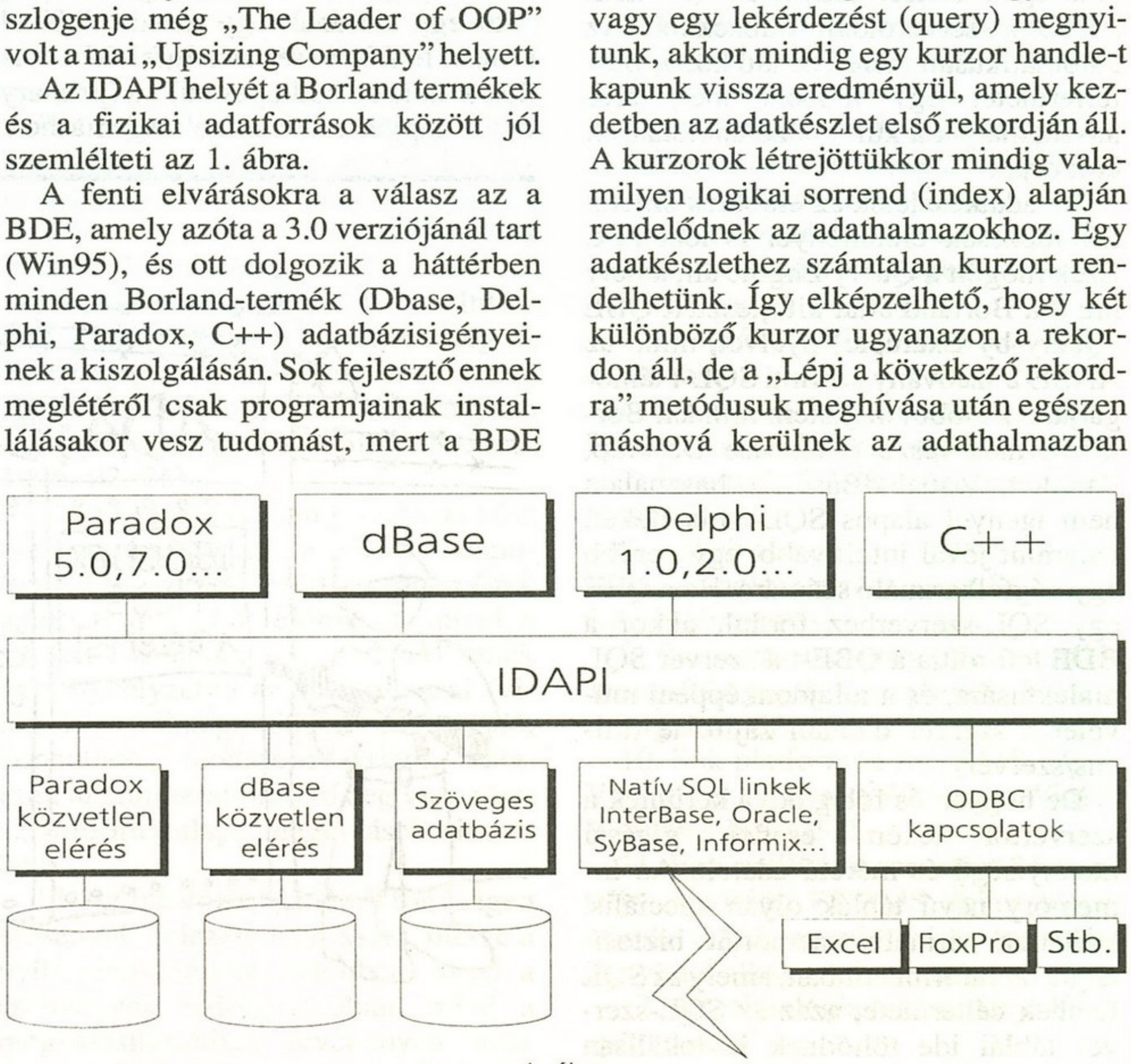
Az IDAPI helyét a Borland termékek és a fizikai adatforrások között jól szemlélteti az 1. ábra.

A fenti elvárásokra a válasz az a BDE, amely azóta a 3.0 verziójánál tart (Win95), és ott dolgozik a háttérben minden Borland-termék (Dbase, Delphi, Paradox, C++) adatbázisigényeinek a kiszolgálásán. Sok fejlesztő ennek meglétéről csak programjainak installálásakor vesz tudomást, mert a BDE DLL-jei nélkül programja halott felhasználói interfész lenne.

Ezek után lássuk, hogy milyen eszközökkel sikerült a fenti, talán maximalistának tűnő igényeket a BDE-nek kielégítenie. Ennek áttekintésére szolgál a 2. ábra.

A első pontban elvárt általános adatelérés biztosítása miatt a BDE az SQL terminológiában használatos „tábla", illetve „kurzor" fogalmakra épít. Tehát egy Dbase-állomány az IDAPI-n keresztül nézve egy Oracle-táblához sok szempontból hasonlóan viselkedik. A táblákat vagy a lekérdezések eredményét képező rekordokat (közös nevén dataset vagy adatkészlet, adathalmaz) a kurzorobjektumokon keresztül érhetjük el. A kurzor címezhető elérést biztosít, de egyszerre mindig csak egy rekordhoz.

Az összes adatmanipuláció (insert, delete, update, fetch) és adatnavigáció (pozicionálás, keresés) kurzorokon keresztül valósul meg. Amikor egy táblát vagy egy lekérdezést (query) megnyitunk, akkor mindig egy kurzor handle-t kapunk vissza eredményül, amely kezdetben az adatkészlet első rekordján áll. A kurzorok létrejöttükkor mindig valamilyen logikai sorrend (index) alapján rendelődnek az adathalmazokhoz. Egy adatkészlethez számtalan kurzort rendelhetünk. Így elképzelhető, hogy két különböző kurzor ugyanazon a rekordon áll, de a „Lépj a következő rekordra" metódusuk meghívása után egészen máshová kerülnek az adathalmazban



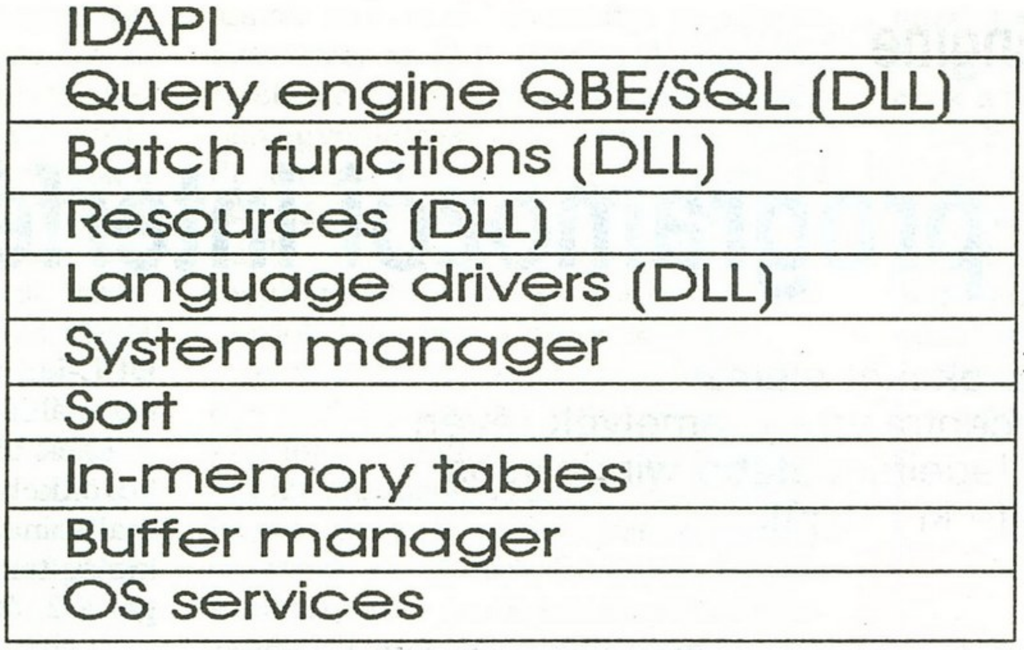
1. ábra

(ha különböző indexek alapján nyitottuk meg őket). Érdekes lehetőség a kurzorok szinkronizációja, amikor az egyik kurzor a másik által mutatott rekordra ugrik. A kurzorokhoz kapcsolódik a könyvjelző (bookmark) fogalma is. Ennek segítségével a kurzor valamikori pozícióját tárolhatjuk el későbbi visszatérés céljából. Erre azért van szükség, mert az SQL-tábláktól idegen a közvetlen rekordsorszámra való hivatkozás.

A ún. „kapcsolt kurzorok" (linked cursors) segítségével pedig megdöbbentően gyors adatnavigációra nyílik lehetőség, pl. master-detail kapcsolatban lévő táblákban. Akár több táblát is összekapcsolhatunk ilyen módon kódmentes programozással, és az eredmény: lenyűgöző sebesség. A különböző driverek kurzorjai által mutatott rekordok egy puffermenedzser által kezelt memóriaterületen vannak. Így a pufferelés miatt az adatmanipuláló utasítások véglegesítése előtt még van lehetőség a cancel műveletre (ez nem azonos a szerveroldali rollbackkel). Az automatikusan méretezendő közös pufferterületet egy időben, megosztva használhatják a különböző adatbázisok kurzorjai.

Az adatkészletek az előbbiek szerint lekérdezések eredményei is lehetnek. Ezek mögött a Query Engine áll, amely mind a Borland által kifejlesztett QBE (Query by Example) nyelvet, mind az ANSI92 szabvány szerinti SQL-t támogatja. Az előbbi majdnem minden Borlandtermék része (Database Desktop, Paradox, Visual dBase...), használata nem igényel alapos SQL-ismereteket, valamint jóval intuitívabb, egyszerűbb egy végfelhasználó számára. Ha a QBE egy SQL-szerverhez fordul, akkor a BDE lefordítja a QBE-t a szerver SQL dialektusára, és a tulajdonképpeni művelet a szerver oldalán zajlik le (kliens/szerver).

De hogyan és főleg hová kerülnek a szervertől lekért, esetleg gigászi mennyiségű és méretű adatok. Az Inmemory nevű táblák olyan speciális, korlátlan virtuális memóriát biztosítanak táblaformátumban, amely az SQL fetchek célterülete, azaz az SQL-szerver táblái ide töltődnek le lokálisan (caching). De ezeket a memóriatáblákat használja az a Sort-engine is, amely a Quick-sort algoritmus nem rekurzív (iteratív) kifejtése elvén működik, és a Query Engine egyik legfontosabb alkotórésze. A batch-funkciók teszik lehetővé — többek között — a táblák adataikkal együtt történő áthelyezését egyik formátumból a másikba (pl.: Oracle—>Paradox, Dbase—>InterBase). A megfelelő típuskonverziók automatikusan végrehajtódnak. Így egyetlen utasítással egy egész tábla átméretezhető lefelé vagy felfelé. Ezenkívül külön batch-funkciók szolgálják például egy táblának egy másik táblával történő felfrissítését. A megfelelő rekordok a táblák elsődleges kulcsa (primary key) alapján rendelődnek egymáshoz.

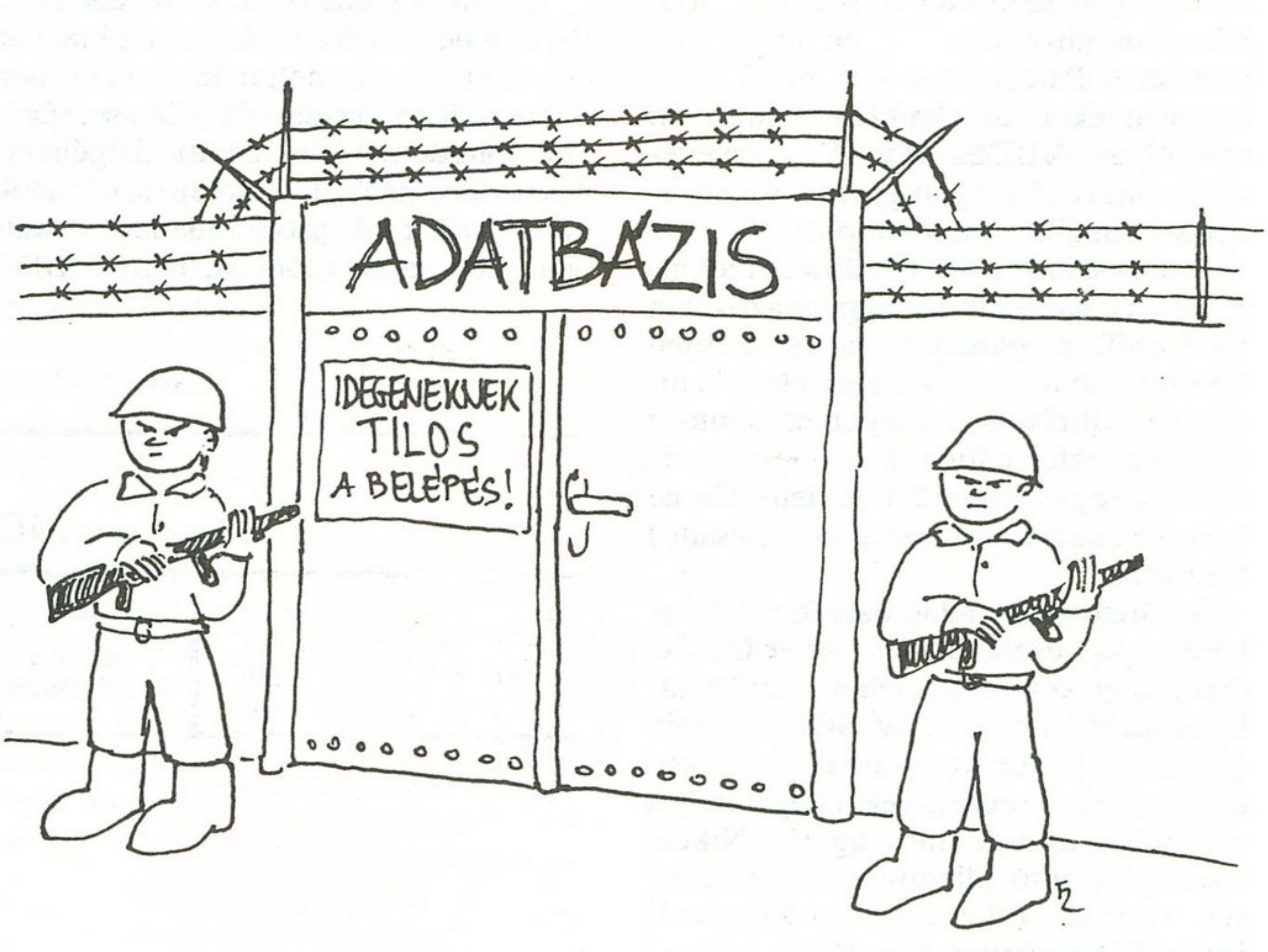


2. ábra

Az SQL terminológiában az elsődleges kulcs megléte kitüntetett fontosságú.

A BDE hordozhatóságát szolgálja az ún. OS services modul, amely magára vállalja az operációs rendszerrel történő alacsony szintű kapcsolattartást (I/O, OS-szintű memóriakezelés, hálózatelérés); ezzel mintegy elszigetelve a BDE-t a konkrét operációs rendszertől. Így vált lehetségessé pl. a Win95-ös BDE-verzió gyors elkészítése, hiszen elegendő volt jobbára csak ennek az összetevőnek az átírása. Számunkra különös jelentőséggel bír a BDE nyelvi hordozhatósága. Ezt a legkülönfélébb karakterkészletek, ill. nyelvi meghajtók (az indexekben megjelenő ábécésorrend) támogatásával érték el. Igazán meglepő, hogy a BDE konfigurálásakor háromféle magyar nyelvi meghajtó közül válogathatunk, de megtalálhatjuk a koreai, héber vagy kínai verziókat is. Szintén a nyelvfüggetlenséget szolgálják a DLL-ekben elhelyezett hibaüzenetek. Ezeket az erőforrásokat, amelyek többnyire egyszerű stringtáblák, pl. a Borland Resource Workshop programjával minden nehézség nélkül magyaríthatjuk.

Naszádi Gábor



Tous les trois

Háromszor tizenöt érv

Márciusi számunkban debütált — a Netware kontra Windows NT pengeváltással — az a rendhagyó forma, amelynek keretében egymondatos érvek feszülnek egymásnak egy-egy termék(család) erényeit summázandó. Az akkori 25-25 érvet most 15-15-re redukáltuk, a résztvevők számát viszont a piaci pozíciók szem előtt tartásával most háromra emeltük. Ennek a megoldásnak sajátságos játékszabálya, hogy a résztvevők csak a lap megjelenésekor ismerhetik meg piaci vetélytársaik álláspontját. Az érintett cégek vezető szakemberei ismét vállalták a nyilvános érvelés nem kevés kockázatot rejtő kihívását: az Informix mellett Sándor Gábor, az Oracle „szószékén" Mohay Tamás érvel, míg a Sybase fogadott prókátora Czuprik Zoltán.

Informix

1. Az első Unix-alapú relációs adatbáziskezelő rendszer (RDBMS).

2. Az első Unix-alapú negyedik generációs programnyelv.

3. Az első Unix-alapú OLTP szerverarchitektúra.

4. A legtöbb TPC (tranzakció-feldolgozási teljesítmények mérésére szolgáló) benchmark-eredményt az Informix publikálta: azonos hardverkörnyezet, illetve applikáció tesztelése esetén az OnLine RDBMS rendre jobbnak bizonyul.

5. Az első párhuzamos adatbázis-architektúra (DSA — dinamikusan méretezhető architektúra), amely egyszerre fedi le az egyetlen processzorral rendelkező, a szimmetrikus multiprocesszoros, a klaszteres és a nagyban párhuzamos (MPP) számítógépeket.

6. Az első objektumrelációs adatbáziskezelő rendszer (Illustra ORDBMS): ezen a területen piaci elemzők szerint az Informix jelentős (mintegy másfélkét éves) előnyre tett szert.

7. A Unix-alapú RDBMS-piacon az installált licencek száma (több mint 900 000) alapján az Informix az első, 34,7%-os aránnyal (IDC, 1995).

8. Az installált licencek száma alapján a legtöbb RDBMS-alapú kliens/szerver szoftvert (26,2 százalékot) az Informix értékesítette (DBMS Magazine).

9. Szintén az installált licencek számát tekintve a legtöbb adatbáziskezeléshez kapcsolódó alkalmazásfejlesztő eszköz (31,2%) Informix-termék (DBMS Magazine).

10. Az elmúlt másfél évben az Informix a leggyorsabban növekedő cég az RDBMS-piacon, az elmúlt öt év adatai alapján a tőzsdén forgalmazott legjobb high-tech cég (The Wall Street Journal, 1996. 02. 26).

11. Az Informix nem forgalmaz kész alkalmazásokat, csak a DBMS technológiára koncentrál, ezért a partnercégek gyakran részesítik előnyben, mivel a végfelhasználónál nem kerülhetnek versenyhelyzetbe az Informixszal (világszerte mintegy 5000 viszonteladó, konzultáns, rendszerintegrátor, valamint alkalmazásfejlesztő cég vesz részt az Informix-alapú megoldások sikerében).

12. Az Informix kezdettől fogva igen erősen elkötelezett a Unixszal, illetve a nyílt rendszerekkel szemben, mind a szabványok kidolgozásában, mind a megvalósításukban tevékenyen vesz részt (ANSI SQL-89, ANSI SQL-92, X/Open NLS, X/Open XA stb.), ezenfelül a de facto szabványokat is beépíti termékeibe (például ODBC, DRDA, DCE).

13. Az első RDBMS, amely az amerikai kormány, illetve az Európai Unió biztonsági (secure) szabványainak eleget tett (NCSC B1, C2, ill. ITSEC), és ezt alapszolgáltatásként (C2), illetve alternatívaként (B1), de felár nélkül nyújtja.

14. Az Informix 41 országban rendelkezik helyi képviselettel, az ügyfeleket 8 regionális vevőszolgálati központ látja el, a szolgáltatások (support, oktatás, tanácsadás) csúcsminőségét 9 ISO-tanúsítvány jelzi, a minőségi igények kielégítésére a nagy hardver- és szoftverszolgáltatókkal közös Competence Centereket hozott létre.

15. Az Informix pénzügyileg stabil cég, saját bevételeiből fejleszt, hosszú távú együttműködési projektjei és vezető technológiája révén minden valószínűség szerint öt év múlva is a legjobbak között találjuk majd.

Oracle

1. Kiváló TPC-C benchmark-eredmények mutatják az OLTP képességeit (V7.3).

2. Kiváló TPC-D benchmark a döntéstámogató alkalmazások számára (Data Warehouse — V7.3)

3. Skálázható, ki tudja használni a rendelkezésre álló gépet, akár SMP, akár MPP, akár cluster.

4. Többféle adattípust tud integráltan kezelni (relációs, szöveges, video — V7.3 Universal Server).

5. Integrált, grafikus adminisztrációs eszközök.

6. Komplett fejlesztőeszköz-család: Power Object, Developer/2000.

7. Integrált CASE-környezet: Designer/2000.

8. Web-támogatás: Web-listener, Web—Oracle interfész (Java, PL/SQL), CASE-támogatás.

9. Biztonság (security), felhasználóazonosítás, jogok, szerepkörök, hálózati forgalom titkosítása.

10. Sok platformon fut: VMS, Unix, VM, MVS, Netware, Windows, Win95, Windows NT, OS/2.

11. Egy szállítótól szerezhető be egy teljes rendszer: RDBMS, fejlesztőeszközök, CASE, gazdasági és termelésirányítási alkalmazáscsomag.

12. A technológia hosszú ideje bizonyított már.

13. Stabil szállító cég.

14. Nagyon sok partnercég támogatja: fejlesztőeszközök, alkalmazások kaphatók hozzá.

15. Support, support, support: a helyi központok saját rendszereiken kívül aktívan használják a globális (USA, Anglia, Ausztrália) központok gépeit, alkalmazásait (és ottani kollégáikat) a problémák megoldására, világszerte 2000 supportos dolgozik, igen erős a magyar support-szervezet is.

Sybase

1. Kiváló teljesítmény/ár viszony az erőforrások hatékony kihasználása következtében (többfonalas architektúra, csekély memóriaigény, optimális SMP-kihasználás).

2. A Sybase mentési és tranzakciókezelési rendszere nagyfokú adatintegritást, adatbiztonságot, helyreállíthatóságot biztosít, és lehetővé teszi a 24 órás folyamatos működést a kritikus adatbázis-karbantartó műveletek alatt is.

3. A felhasználó számára kedvező a termékkövetési rendszer: alacsony support-díjak, amelyek tartalmazzák az upgrade-et az új verziókra — „nagy" verzióváltás esetén is.

4. Skálázhatóság, az egyedi PC-től a több ezer felhasználót kiszolgáló szerverekig.

5. Nagyvállalati, illetve világméretű rendszerek is létrehozhatók a Replication Server segítségével, amely az adatforrások összekapcsolását és a tranzakciók lebonyolítását akkor is lehetővé teszi, ha a távoli adatkapcsolat átmenetileg nem működik.

6. Az EnterpriseConnect middleware termékcsalád segítségével különböző gyártóktól származó különböző jellegű adatforrások egységes rendszerbe integrálhatók, az egyszerű szövegfájloktól kezdve a nem Sybase adatbáziskezelőkig.

7. Értékmegőrző beruházás: a korábban „szigetszerűen" létrehozott heterogén adatbázisok az egységes Sybase-alapú nagyvállalati rendszerben tovább működnek, ilyenkor a felhasználók számára az összes adatbáziskezelő Sybase-szervernek látszik.

8. Valódi nyíltság mind a kliens-, mind a szerveroldalon: a Sybase a felhasználó által szabadon programozható interfészt ad a kliens- és a szerveroldalon egyaránt, ez lehetővé teszi a felhasználó egyedi problémáinak megoldását, illetve más gyártók termékeinek rugalmas illesztését.

9. Az alkalmazások teljes életciklusának támogatása: S-Designor (adatfolyam-modellezés, logikai és fizikai adatbázistervezés, adatbázis-generálás, alkalmazás-prototípus készítése; korábban készített alkalmazás modelljének előállítása), PowerBuilder (kliensoldali alkalmazásfejlesztés és verziókövetés), SQL-Server Manager (adatbázis-létrehozás és -karbantartás — rendszeradminisztrátori feladatok), SQL-Debug (szerverprogramok és SQL scriptek belövése), SQL-Monitor Server (adatbázishangolás).

10. A Sybase Enterprise Messaging Services (EMS) lehetővé teszi, hogy mobil felhasználók sokaságát kapcsoljuk a nagyvállalati rendszerhez.

11. Data Warehouse-támogatás: a Sybase-IQ a terabájt méretű adatbázisok lekérdezésének hatékony eszköze, amely az OLAP (On-Line Analitical Processing) eszközök számára is lehetővé teszi, hogy a vállalat meglévő relációs adatbázis-adataiból kiindulva sok adattal sokféle analízist végezhessenek — szemben a kizárólag elemzési célokra létrehozott multidimenziós adatbázisokkal, amelyek viszonylag kevés adattal korlátozott számú analízis elvégzésére képesek.

12. Internet-támogatás: a Sybase web.sql kapcsolatot teremt a közismert web-szerverek és a relációs adatbáziskezelők között, így a HTML-lapokra írt SQL utasítások segítségével az adatbázistartalom nemcsak lekérdezhető, hanem tetszőleges relációs adatbázisművelet kezdeményezhető (tehát beszúrás, törlés, módosítás, sőt adatbázis-adminisztráció is), ezáltal komplett alkalmazások is megvalósíthatók — természetesen csak a megfelelő jogosultság esetén.

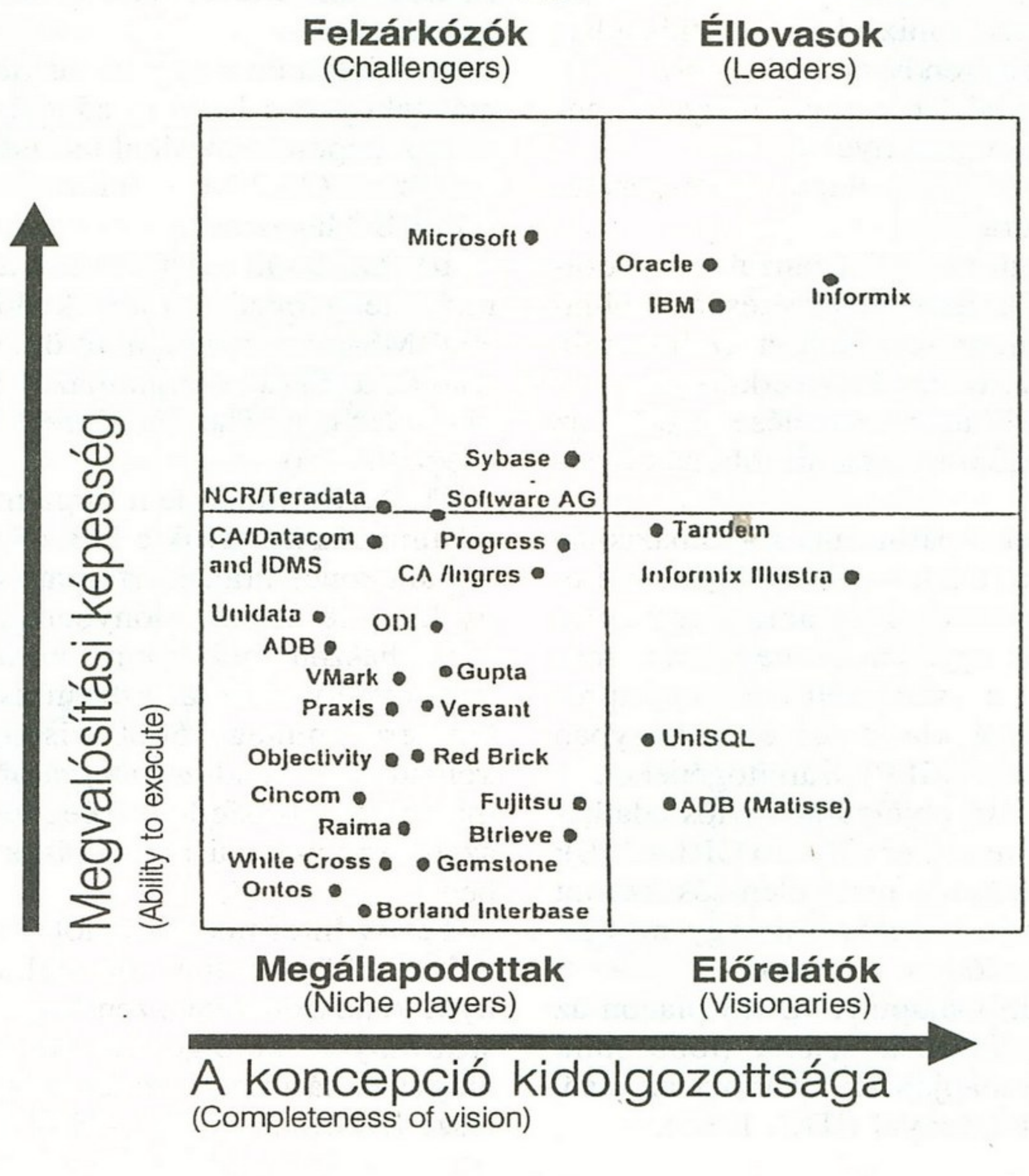
13. A PowerBuilder alkalmazásfejlesztő különleges szolgáltatásai: DataWindow, OLE 2.0, DDE, Internet-hozzáférés, osztott alkalmazások készítése.

14. Az objektumorientált fejlesztési módszerek széles körű támogatása: klienszinten teljes OO-tervezés és programfejlesztés négyféle módszertan támogatásával; szerverszinten pedig az új ObjectConnect middleware programcsalád a relációs adatbáziskezelő objektumorientált használatát teszi lehetővé.

15. A Sybase termékei, valamint támogatásai (technical support, customer service, account management) rendelkeznek ISO 9001 minőségi bizonyítvánnyal.

A DBMS „varázsnégyszöge"

(A Gartner Group 1996. februári elemzése alapján)



OS/2 Warp

Beszél már Ön a számítógépéhez?

**Az IBM tette fel a címben szereplő kérdést sok ezer alkalmazásfejlesztőnek az 1996. április 23-án, Nashville-ben tartott konferencián.
A Merlin legnagyobb szenzációja kétségkívül a beszédfelismerés. Beszéddel lehet majd vezérelni a következő generációs OS/2-vel felszerelt gépet. Ezzel nagyon leegyszerűsödik a böngészés a világhálón, programokat lehet egy-egy szóval indítani.**

Ha valóra válik a Merlin fejlesztőinek terve, akkor majd diktálni is lehet már a számítógépnek — ehhez 10 000 szavas alapszótárat mellékelnek. Magyarul sajnos nem fog érteni a Merlin, szótárat valószínűleg csak angol, amerikai angol, német, spanyol, francia és olasz nyelvhez lehet majd kapni. A beszédfelismerés használatához gyors koprocesszorral rendelkező processzor (minimum 90 MHz-es Pentium) vagy Mwave DSP kártya kell. Azonban a processzorok árának folyamatos esésével ez az otthoni felhasználó számára is elérhetővé válik.

A Java terjedésével egyre több szoftvergyártó jelenti be a Java támogatását. A Microsoft a Windows következő változatába akarja beépíteni, az IBM azonban — megelőzve vetélytársát — a teljes Java-támogatást beépíti a Merlinbe, így valószínűleg ez lesz a második Javát tartalmazó operációs rendszer az AIX után.

Egy másik fontos újítás a TrueType betűkészletek kezelése. Meg nem erősített hírek szerint a WordPro for OS/2 bétaverziója TrueType fontokat jelenített meg. Ezzel jelentősen egyszerűsödik a szövegszerkesztők közötti adatcsere, hiszen elég csak TrueType betűkészletnek rendelkezésre állnia. Ezenkívül a WordPróba beépítik a beszédfelismerési lehetőséget, így tehát lehet majd leveleket diktálni a programnak.

A Merlin kezelői felülete nagyban megváltozik: hasonlítani fog a Windows 95-re. A LaunchPad (Gyorsindító, Kilövőállás) helyére a SmartCenter ke-



rül, ahol nyomon lehet követni a futó programokat, a lemezhasználatot, az időt, a dátumot és még sok fontos dolgot. Az ablakok jobb felső sarkába pedig záró X gomb kerül.

Lotus Smartsuite

Nagyjából a Merlin megjelenésekor fog a boltokba kerülni a Smartsuite legújabb OS/2-es változata is. Ez a fent említett WordPrón kívül tartalmazza még a Freelance Graphics, 1-2-3, Approach és az Organizer programokat. Ezek mindegyike valódi 32-bites alkalmazás, alternatívát nyújtva a Microsoft Office-szal szemben.

Developer's Connection

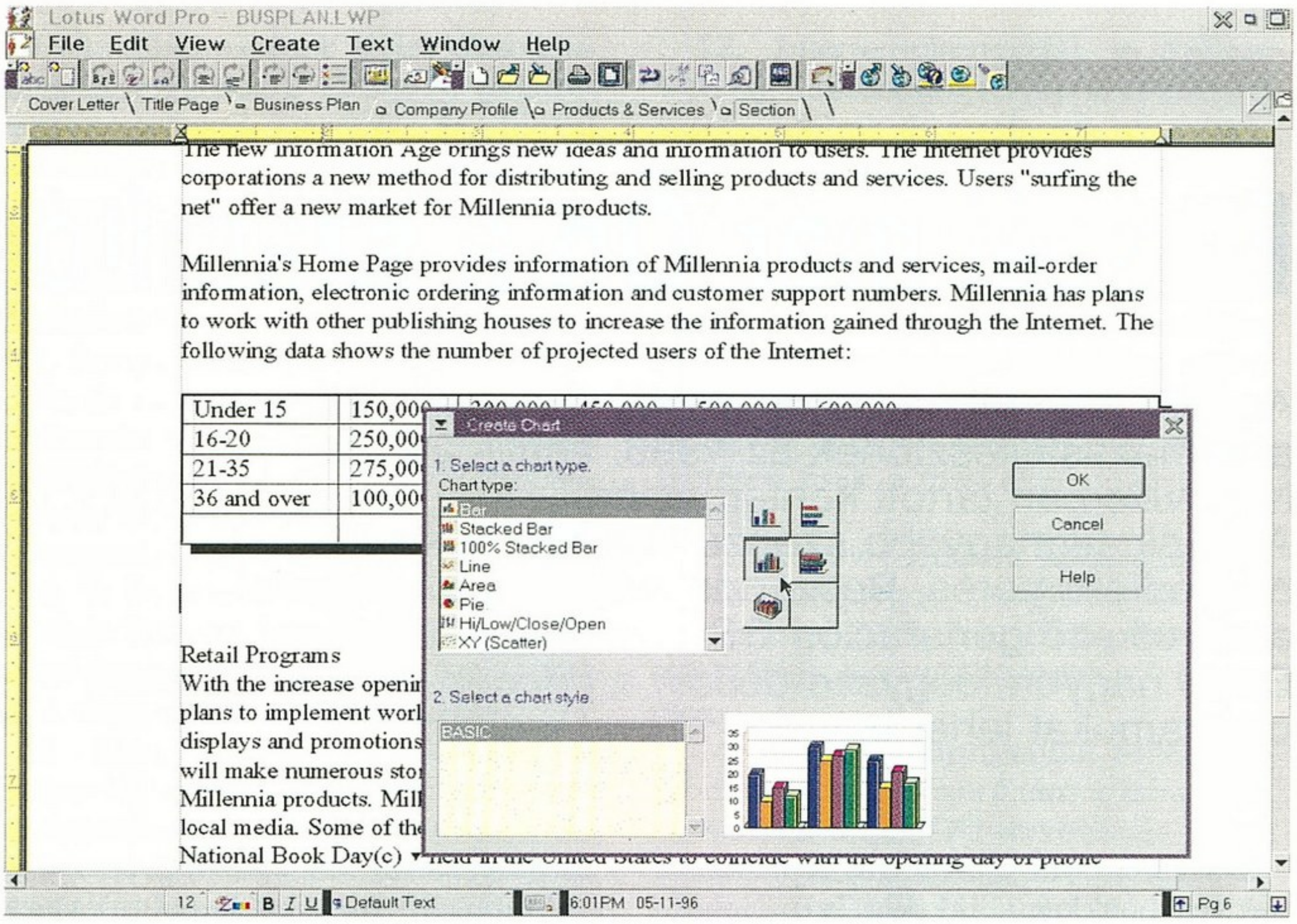
A Merlin mellett más OS/2 hírek is megjelennek az elektronikus világban. Április 18-án megjelent a fejlesztőknek szóló Developer's Connection (DevCon) 10-es verziója. Ebben újdonságként az AIX és OS/2 verziót összevonták. A CD-ken több száz Mbájt információ, eszköz, demó, fejlesztőeszköz, példaprogram található, elsősorban programfejlesztőknek szánva. Az előfizetők negyedévente kapják a frissített CD-ket.

OS/2 Warp Server

Az IBM 1996. május 8-án bejelentette több mint 50 000 db OS/2 Warp Server eladását. Ez a szám jelentősen meghaladja a várakozásokat. Például Karolina (?????) állam egymaga 300 Warp Server licencet várásolt, az Antartica brazil cég pedig 900-at. A Warp Server az OS/2 Warp operációs rendszeren alapuló alkalmazáskiszolgáló. Ezenfelül képes PC LAN gépeket átfogni, és nagy biztonságot igénylő hálózatokat is kiszolgálni. A LAN Server és az OS/2 egybeépítésével az IBM nagyon egyszerűen kezelhető, mégis nagy teherbírású kiszolgálót készített. A Warp Server kis- és nagyvállalatoknak egyaránt megbízható szolgáltatást nyújt. Megkönnyíti a rendszerfelügyeletet, támogatja PostScript fájlok nyomtatását nem PostScript nyomtatókra, és a hálózati protokollok rendkívül széles választékát nyújtja.

BESTeam program

Az IBM Magyarországon is megindította a viszonteladóknak szóló BESTeam programot. Az IBM a program keretében a viszonteladókat információkkal, CD-ROM-okkal látja el, oktatást



WordPro béta — egyelőre még Warp 3.0 alatt

szervez a számukra, megismerteti őket az IBM szoftverekkel. Az így felkészített BESTeam tagok majd szakmai tanácsokat adhatnak a hozzájuk forduló vevők IBM-termékekkel kapcsolatos kérdéseire.

Játékok

Megjelent a közkedvelt Galactic Civilisation nevű játékprogram újabb verziója, amelyben már könnyen lehet hajót, technológiát, idegeneket stb. készíteni. A csak OS/2 alatt futó programról többet a http://www.stardock.com címen lehet megtudni. Lehet jelentkezni a Trials of Battle című, kb. a Doomot és Wing Commandert egyesítő játék június 1-jén megjelenő bétaverziójának tesztelésére — szintén a http://www .stardoc.com címén. Aki inkább a logikai játékokat kedveli, az letöltheti az Entertainment Packot az ftp://hobbes.nmsu.edu/incoming vagy az ftp:// wilmington.net/bmtmicro címről. A fájlnév: entrtn27.zip az amerikai verzió esetén, illetve entrt27g.zip a német verzió esetén.

A csomag klasszikus játékok OS/2-es változatát is tartalmazza (például: Tetris, Othello, Master Mind, Tic Tac Toe stb.).

Vegyes

A Netscape bejelentette, hogy népszerű böngészőprogramját OS/2-es platformon is meg kívánja jelentetni. Konkrét időpontot még nem adtak meg, de a honlapjukon (http://www.netscape.com) lehet friss híreket keresni ezzel kapcsolatban.

Két, több száz Mbájtnyi OS/2-es shareware-t, freeware-t, demóprogramokat és meghajtókat tartalmazó ftp-kiszolgálót tükröznek Magyarországon: a http://xenia.sote.hu/vmirror/os2/ munich/Xenia.html az ftp.leo.org tükrözése, a http://xenia.sote.hu/vmirror/os2/hobbes/HVMindex.univie.html pedig a hobbes.nmsu.edu-é.

Már 1995 augusztusa óta működik az OS/2 és TeamOS/2 levelezési lista. Míg az OS/2 levelezési listán főleg OS/2-vel kapcsolatos tippek, ötletek olvashatók, addig a TeamOS/2 levelezési listán a Team-tagok érintkeznek. Bárki feliratkozhat a listákra a majordomo@lists.bme.hu címre küldött levéllel. A levélben a következő sor szerepeljen: subscribe os2 vagy subscribe teamos2. Az OS/2 levelezési lista eddigi anyagába már három helyen is bele lehet nézni:

- http://xenia.sote.hu/docs/ mailarchive/os2
- http://www.vma.bme.hu/os2/os2
- gopher://kiribati.et.tudelft.nl/ 1os2lista/os2lista.70

A ftp://xenia.sote.hu/incoming/os2/ teamos2/tcpip_v4.zip címről letölthető a TCP/IP 4-es verziójának bétaverziója. A végleges verziót a Merlin fogja tartalmazni.

Az Team OS/2-es levelezési lista [teamos2@lists.bme.hu] májusi anyaga, a comp.os.os2.announce levelei és az IBM (http://www.ibm.com) honlap alapján írta:

Tóth Ferenc

CA-Visual Objects 1.0c

Megújult környezet

A CA-Clipper forrásnyelvével felülről kompatibilis adatbáziskezelő és alkalmazásfejlesztő rendszer legújabb verziója elmozdult a Windows 3.1/3.11 alapról, stabillá vált, felveszi a versenyt más szoftverházak legkorszerűbb fejlesztőeszközeivel.

A CA-VO grafikus felhasználói felülettel rendelkező (GUI), kliens/szerver alkalmazások készítésére kifejlesztett, 4GL Xbase, teljes mértékben objektumorientált programozást (OOP) megteremtő, valódi végrehajtható kódot előállító (Native Code Compiler) program.

Repository-alapú integrált fejlesztői környezet (IDE), előre elkészített osztálykönyvtárak (ClassLib), különféle adatállomány-hozzáférési módok (RDD, SQL, ODBC), vizuális tervezőeszközök (Painters, FormEditor, MenuEditor, DBServerEditor, Entity Browser, Error Browser stb.) segítik a munkát. A különféle alkalmazások közötti kommunikációt teljes körű dinamikus adatcsere (DDE) támogatja.

Előzmények és a közeljövő

A VO 1994 decemberére készült el, és már 1995 januárjában megjelent az 1.0a INTL-verzió, ez került kereskedelmi forgalmazásra. Áprilisra elkészült a forráskódokat tartalmazó Software Development Kit is, majd június végére az 1.0b update, mindkettő CD-n. Egy hónappal később önálló termékként jelent meg a piacon a CA-Visual Objects Lite. Ezt az olcsó változatot a termék üzleti bevezetésére és az OOP-re való átállás elősegítésére szánták. A Lite nem tartalmazza a Watcom SQL adatbáziskezelőt, a CA-RET riportgenerátort és az ODBC DLL-eket. Az év végére vált elérhetővé a legújabb frissítés, a CA-Visual Objects 1.0c. Napjainkban pedig a CA-VO 2.0-s változat bétatesztje kezdődött meg. A kereskedelemben várhatóan ez év nyarán megjelenő verzió bétatesztelésébe magyar fejlesztők is bekapcsolódhatnak. (A magyar felsőoktatásban is oktatják a szoftvert, többek között a Gábor Dénes Műszaki Főiskolán is). A Computer Associates Inc. augusztusban rendezi meg New Orleansban szoftverfejlesztői konferenciáját, a TenhniCont, s itt tervezik bemutatni a CA-VO teljes mértékben 32 bites válltozatát, a CA-Visual Objects 2.0-t. Ezenkívül olyan témák is terítékre kerülnek, mint pl. a CA-VO adatbázisrendszerek tervezése, hogyan készítsünk pufferelt adatablakot, OLE és OCX technológiák, osztály-implementáció, hogyan „debugoljunk", memóriafoglalás, mutatók és típuskényszerítés, többszálas programozás, Oracle adatbázis-hozzáférés CA-Visual Objects 2.0-ban, felhasználói felület tervezése Win95/NT alatt a CA-VO fejlesztők számára stb. Érdemes ez utóbbi témánál megállni egy pillanatra.

Win95/NT: felülettervezési irányelvek

A Win95/NT operációs rendszer újfajta adatközpontú felhasználói interfész tervezését javasolja. Minden fejlesztő, aki lépést akar tartani a fejlődéssel, jó, ha elgondolkodik az általa megvalósított felhasználói felület használhatóságán. A VO-fejlesztők szempontjából az egyik legfontosabb kérdés az egydokumentum-interfész (SDI) és a többdokumentum-interfész (MDI) közötti választás. Az SDI esetében az alkalmazás egyidejűleg az adatnak csak egyik nézetét engedi meg. Ha két dokumentumot akar egyszerre nézni, akkor a programnak két példányát kell megnyitnia. Az MDI esetében több dokumentum is megnyitható, illetve egy dokumentumnak több nézete is látható egyszerre az ablakban. Ekkor az ablak mintegy konténerként tartalmazza a különféle adatokat.

A legtöbb CA-VO alkalmazáshoz az MDI tűnik a megfelelő választásnak. Ekkor a felhasználó egyszerre több táblát is nézhet, vagy egy táblát, több módon. A Win95/NT irányelvek nem befolyásolják ezt a vélekedést, azonban rámutatnak az MDI korlátaira az adatközpontú környezetben.

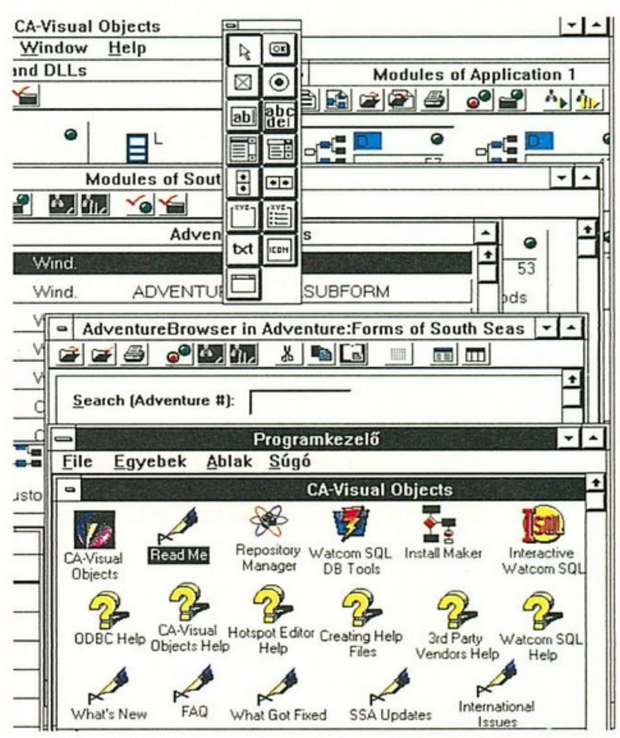
Az egyik probléma az adat és az ablak közötti kapcsolat. A főablak csupán egyik konténere az al-ablakoknak, de nem az adatoknak, melyeket az ablakok tartalmaznak. Ha a főablakot bezárjuk, majd újra kinyitjuk, azt várjuk, hogy ugyanazokat az al-ablakokat fogjuk látni megint, ugyanott és ugyanazokkal az adatokkal. Az alkalmazás nyomon tudja követni, hogy mi történik az ablakokkal, de az adatok időközbeni változásait nincs módjában követni. Mi van akkor például, ha egy al-ablak által megjelenített adatot időközben letörlünk?

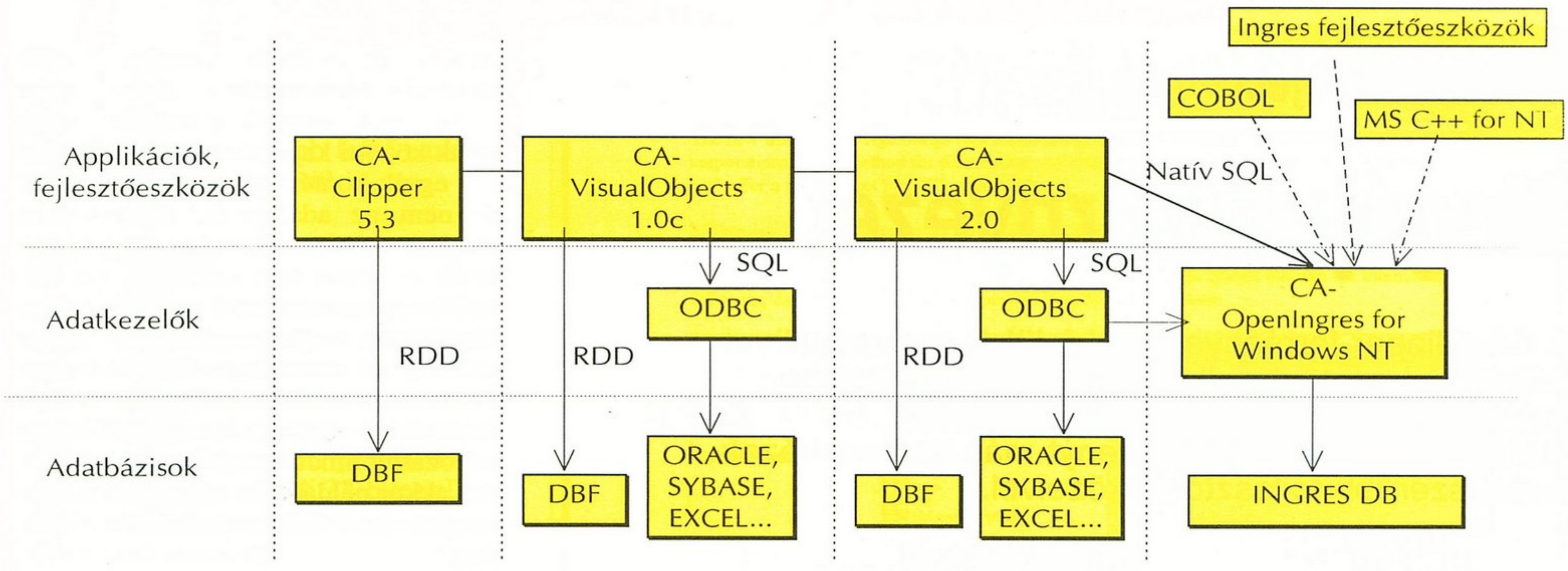
Továbbá az ablakokkal való munka sem mindig konzisztens. Az első fájl megnyitásához a felhasználó egy mappából vagy a Win95/NT Dokumentumok menüjéből választhat. További dokumentumok megnyitásához már a felhasználónak az alkalmazás interfészét kell használnia. Ezért a kezdők számára az MDI nagyobb kihívást jelent.

Az irányelvek az SDI támogatása helyett inkább három lehetőséget ajánlanak a több dokumentum megtekintéséhez. Az első a munkaasztal. A Win95/NT-ben a felhasználó kiválaszthat egy objektumot (egy ikont), és onnan egy al-ablakba léphet. Azonban ez — ellentétben a mappákkal — összefügg a főablakkal, és így ugyanolyan kezelést igényel, mint az MDI.

A második lehetőség a munkafüzet. Ez egy egyszerű ablak, amely sok oldalt tartalmaz, és minden oldalt egy kis füllel érhetünk el.

A harmadik választási lehetőség a projekt. Ez megintcsak az MDI-hez hasonlít, azt kivéve, hogy a főablak, ha egyáltalán van olyan nem viselkedik konténerként. Az al-ablakok közvetlenül a munkaasztalon helyezkednek el,





és minden al-ablaknak megvan a saját ikonja a Win95/NT menüsávjában (taskbar). A főablak tartalmazhatja a főmenüt, és biztosíthatja az ablakok kezelését, vagy az egyes al-ablakoknak is lehetnek független menüi és eszközsorai. Ezt az interfészt néhány fejlesztői platformon használják, ilyen például a Microsoft Visual Basic vagy a Borland Delphi.

Ez a három alternatíva nem standard. Pusztán csak egy adatcentrikus környezet felhasználói interfészének felépítési javaslata. Ezek tulajdonképpen nem léteznek a Windowsban — például nincs API-hívás egy munkafüzet megnyitására, és azt sem jelentik, hogy szükségszerűen és kizárólag ezeket kellene használni. Tervezhetünk akár olyan interfészt is, amely kombinálja a kettőt, vagy akár mindhárom elemeit tartalmazza. Röviden az út általános irányát adják meg, és nem egy adott járművet.

Műveletalapú alkalmazások

Az Microsoft útmutatóinak hiányossága, hogy nem teljesen fedik le a műveletalapú alkalmazások szükségleteit. A hangsúly a szövegszerkesztőkön, táblázatkezelőkön stb. van, ahol a dokumentum betöltődik a memóriába, azt ott megszerkesztjük, majd visszaírjuk a lemezre. Ez ellentétben áll a tipikus CA-VO-alkalmazásokkal, ahol a felhasználó a rekordok közötti navigálással és a műveletek feladásával foglalkozik.

Sok CA-VO-alkalmazásban a felhasználó nem találkozik fájlokkal, osztályokkal vagy könyvtárakkal. Ha az alkalmazás például egy vevő adataival dolgozik, akkor majd megnyílik a „vevő adatai" tartalmú fájl is, amikor szükséges. A felhasználó nem foglalkozik sem fájlok megnyitásával, sem elmentésével, mivel ezek a műveletek egyetlen utasítással automatikusan megtörténnek, így a fájlnyitás, fájlmentés és fájlzárás utasítások értelmetlenek számára. Az irányelvek ezen a ponton még kimunkálatlanok. Ez nem jelenti azt, hogy az irányelvek nem megfelelőek a műveletalapú alkalmazásokhoz, de figyelembe kell venni azok határait. Sok alkalmazás azért mehet tönkre, mert a fejlesztők megpróbálják a dokumentumalapú programok szabványát alkalmazni rájuk. A legelemibb hiba a fájlkezelés beiktatása a menükbe.

Más vonatkozásban azonban az irányelvek jól alkalmazhatók adatalapú alkalmazásokhoz, és kimondottan jól megfelelnek a CA-VO-fejlesztőknek. Természetesen a legmegfelelőbb interfész megtervezéséhez nem elég az irányelvek követése. Továbbra is kell a kreatív kísérletezgetés, a tesztelés.

Újdonságok

A CA-VO 1.0c patch terjedelme is jelzi, hogy szinte teljesen új szoftvert fejlesztettek ki a CA programozói.

Még mindig 16 bites az alkalmazás, de a Win95/NT alatti fejlesztések biztosítására valamennyi rendszer-DLL-t újraírták. Az új verzióban kijavították a korábbi hibákat. A fordítási és futási sebesség jelentősen nőtt, gyors fejlesztéséhez már 8 MB memória is elegendő.

A menüelnevezések egységesek lettek, Win95/NT-hez illeszkedik a fájlkezelés, és korrektül működik a Debugger és a Call Stack is. A CA-Visual Objects 1.0c kétbájtos karaktertárolása révén megjeleníti a legkülönfélébb ékezetes betűket is. Az SQL-ben használható tizedesvessző és decimális elválasztójel is, mint pl. az ezres pont.

A megújult CA-VO teszteredményei szerint a fejlesztők alapos munkát végeztek, még a korábban szórványosan előforduló hibákat is megjavították. Többszöri futtatás és tesztelés során sem fordult elő az előző változatnál sajnos gyakori általános védelmi hiba (General Protection Failure).

Jó néhány osztály kibővült új tulajdonságokkal és metódusokkal. Ilyen például a DataBrowser, DbServer, FieldSpec, DbFileSpec, HelpDisplay, Menu, Printer, ToolBar stb. Az új DbFileSpecben lényegesen több információ érhető el az adattáblákról, indexekről és az adatbázismeghajtókról.

Az új termék, a CA-Visual Objects 1.0c nagy előnye az is, hogy egységes forrásból kínál egymáshoz jól kapcsolódó eszközöket, így a CA-Visual Objects klienseket kiszolgáló CA-OpenIngres/Desktop és a CA-OpenIngres for Windows NT adatbáziskezelő szerverprogramot is.

CA-OpenIngres

A CA-OpenIngres/Desktop a jól ismert nagygépes CA-Ingres adatbáziskezelő PC-s változata. Az SQL-alapú relációs adatbáziskezelő szerverprogram, mely teljes Ansi92 SQL-támogatást élvez, egyfelhasználós, többfeladatos adatbáziskezelő. Kliensprogramjai: az SQLTalk az SQL felhasználói interfésze, az SQLEdit konfigurációs és segédprogramokat tartalmaz, a Dberror egy adatbázishibákat leíró programkönyvtár.

A CA-OpenIngres for Windows NT egy SQL-alapú relációs adatbáziskezelő szerverprogram Windows NT-hez. Teljes Ansi92 SQL-támogatású, többfelhasználós, többfeladatos adatbáziskezelő. Kliensprogramjai: Visual DBA, grafikus felhasználói interfész, adatbázis-adminisztrációs és egyéb segédprogramok.

Konkolyné Bihari Zita

Okosodjunk egy kicsit

PC Suli

Gyártó: Zombie Soft
Típus: Újrakezdő, középhaladó angolnyelv-oktató interaktív CD-ROM
Minimális/ajánlott hardverkövetelmények: Windows-kompatibilis hangkártya, legalább dupla sebességű CD-olvasó
Szoftverkövetelmények: Windows 3.1, Windows 95 vagy Windows NT
Várható ár: kb. 8000 Ft + áfa

A magyar Zombie Soft cég második nyelvtanító CD-je, amely nagyságrendekkel jobb kiállítású és szerkesztésű, mint elődje, az „Angol kezdő CD". Míg utóbbi csak egy viszonylag körülményes DOS-os keretrendszerrel működött, addig a mostani középhaladó a legkorszerűbb windowsos vezérlésű, mikrofonkezeléssel. Megjegyezzük, hogy a kezdő DOS-os változat „windowsosítása" is megtörtént azóta, egy upgrade-lemezt lehet kérni a már meglévő CD mellé. A csapat a „Haladó angol nyelvi CD"-t 1996 augusztusára ígéri.

Mint a program kissé hosszúra sikeredett címe is mutatja, a CD két tudásszintre tagolt, aktuális tudásszintfelmérő teszteket is tartalmaz. Mindjárt az elején alávethetjük magunkat egy felmérésnek, így az itt elért eredménynek megfelelően vághatunk neki az újrakezdő egyszerűbb tudásanyagnak vagy a középhaladók leckéinek. Ha szorgalmasan átküzdöttük magunkat a CD-n, egy zárótesztben adhatunk számot az elsajátítottakról. A teszt sikeres elvégzése sajnos még nem jogosít fel semmire, de biztató előjel az állami, középiskolai vagy egyetemi nyelvvizsga előtt állóknak.

Az újrakezdő és középhaladó szinten összesen 20 témakört járhatunk körül. Egy témakör 4 részből áll: olvasmány, témaköri gyakorlat, nyelvtani tananyag, nyelvtani gyakorlatok + szószedet.

Eredeti anyanyelvi hanganyagot kapunk. A felolvasás sebessége is beállítható, ha úgy érezzük, hogy túl jók vagyunk, pörgessük fel egy kicsit a beszéd ritmusát! A nyelvtanulás egyik legnagyobb rákfenéje, hogy az élő beszéd (például egy angol nyelvű film szövege) a tanuló számára még túl gyors. Aki csoportosan tanul, és a gyengébb beszédértésű, az ilyen szöveget hallva rendszeresen lemarad, elveszíti a fonalat. Kudarcélménye előbb-utóbb a csoportból való lemorzsolódáshoz vezet. A CD révén mindenki saját képességének megfelelő beszédsebességet állíthat be, és azt később tanulmányai előrehaladtával normál szintre emelheti, esetleg azon túl is felgyorsíthatja (ha, mondjuk, tőzsdei alkusznak készül).

A CD egyébként ügyes, mintegy 6400 szavas hangos szótárral, az összes rendhagyó igét felsoroló táblázattal, nyelvtani súgóval segíti a tanulást. A zárótesztben már a beszédértést is vizsgálja a program. Ismeretlen, hosszabb szöveget kétszer elismétel a gép, majd a harmadszori meghallgatással egyidejűleg igennel, nemmel, nem tudnival kell válaszolni a témára vonatkozó kérdésekre.

Ízelítőül néhány témakör: egyén, család, lakás, otthon, munka, orvos, autó, könyv, mozi, tv, utazás, repülés, munkavállalás, álláskeresés, gyermeknevelés, Magyarország, Budapest stb.

Szavak, kifejezések, sőt komplett mondatok mikrofonba mondásával az ember összevetheti saját kiejtését a CD-n található angol anyanyelvű narrátoréval, és addig sulykolhatja, ízlelgetheti a saját és más kiejtését, intonációját, amíg a kettő legjobban fedi egymást. A szégyenlősebb tanulók négy fal között egymaguk akár százszor is elismételhetik ugyanazt, a PC Suli tanárai nem lankadnak, birkatürelműek. Csupán állhatatosnak és céltudatosnak kell lennünk, valamint magunkkal szemben nem elnézőnek, hanem szigorúnak. De hiszen erről szól a nyelvtanulás. Nem is csak a nyelvtanulás...

More IQ Test Vol. II

Gyártó: Virtual Entertainment, Inc.
Típus: Intelligenciateszt és egyéb agytornák
Várható ár: 5000 Ft + áfa

Minimális/ajánlott hardverkövetelmények PC: 486-os processzor, 4/8 MB RAM, SVGA monitor, 256 szín, dupla sebességű CD-olvasó, opcionálisan hangkártya. Macintosh: ...030 vagy fejlettebb processzor, 4/8 MB RAM, 13 collos vagy nagyobb monitor 256 színnel, CD-olvasó

Minimális/ajánlott szoftverkövetelmények PC: MS DOS 5.x, 6.x, MS Windows 3.1. Mac: System 7.0

Korábban már ismertettük ennek a CD-nek az első részét. Itt is elmondható, hogy a szerzők okultak az előző rész gyengéiből, és alaposan felülmúlták előző kiadványukat. A More IQ Test szebb és jobb. A feladatok nem mindig azonosak, hanem a program variálja őket. Így többször is kitöltheti az ember az IQ-tesztet, de kétségtelen, hogy másodszor vagy sokadszor már nem éri meglepetésként egy adott kérdéstípus, így óhatatlanul javulni fog az eredménye. Ha valaki tehát hiteles képet akar kapni saját vagy gyermeke(i) intelligenciájáról, problémamegoldó készségéről és találékonyságáról, csakis az első nekirugaszkodásra elért eredményt tekintse mérvadónak. A program tárolja a tesztet elvégző emberek adatait, és egy — a játékokból megszokott „Hall of Fame" — dicsőségtáblán rangsorban fel is tünteti eredményüket.

Nem angol anyanyelvűek számára kedvező változás, hogy magának az IQ-tesztnek a megoldásához alig kell tudni angolul. Így a 60 feladatra adott 45 perc elegendőnek bizonyulhat bárkinek, nem kell percekig bíbelődni a feladat megfogalmazásának megértésével.

A klasszikus IQ-teszten kívül szépen, grafikusan megvalósított öt további fejtörő is található a CD-n, és az adatok ezekben is újra meg újra mások. A gép méri a megfejtésre felhasznált időt. Lehet versenyszerűen is értékelni a teljesítményeket. Az öt agytorna: origami, kombinációs zár kinyitása, toronyépítés összeillő elemekből, 6 lapjára kiterített kockából felismerni az összeragasztottat, különböző tárgyakkal kiegyensúlyozott két mérleg alapján kitalálni a harmadik mérleg második serpenyőjének tartalmát a hat megadott lehetőségből.

A CD igyekszik tudományosan és hétköznapi módon is megmagyarázni, hogy mi az intelligencia. Bemutatja, hogy az intelligencia és a hülyeség között mennyi finom árnyalat és fokozat helyezkedik el, és grafikonon is megnézhetjük ezek előfordulási valószínűségét, eloszlását földünk civilizált népességében.

Az intelligenciamérés kényes és bonyolult dolog. Az ember jobban szereti maga, otthon, mások szeme elől elzárva megtudni, mi is van (vagy mi nincs) a fejében. Ha azután elégedetlen akár a

programmal, akár az elért eredménnyel, csak kattintson rá az Uninstall ikonra...

Novlib 3

Gyártó: Blink, Inc.
Típus: Novell hálózatkezelő függvények programozóknak
Minimális/ajánlott szoftverkövetelmények: Novell NetWare 2.10/3.x, MS DOS 3.0, támogatott nyelvi fordító
Várható ár: kb. 39 000 Ft + áfa

A Blink, Inc. cég Blinker nevű szenzációs programszerkesztőjével robbant be a köztudatba, még 3-4 évvel ezelőtt. Főleg a Clipper-programozók respektálták, hiszen a Clipperhez nem csomagoltak „világbajnok" linkereket. Kezdetben a Plink86-ot, majd az Rtlinket kapták, de ezek a linkerek lassúak és körülményesek voltak, csak vérrel-verejtékkel vehetők rá az overlay technikára. (Általában az azonos nevű önálló termékek korlátozott, kiherélt változatai szerepeltek. Az 5.3-as Clipper-verzióban már a Blinker könnyített kiadását találni.)

Nos, ez a jó nevű cég kiadott egy teljes körű hálózatos függvénykönyvtárt, a Novlibet. Miért érdemes rá odafigyelni? Több mint 450 függvény tesz lehetővé a következőkhöz hasonló funkciókat:

— Előzetes login nélkül lekérdezhetők hálózati paraméterek.

— Nem hagyja figyelmen kívül a Novell mindenkor beállított illetékességi rendszerét és hozzáférési jogait, az egyes csoportbeállításokat.

— Programból állítható hozzáférési jog, de csak akkor, ha a programot éppen használónak ehhez joga is van.

— Munkaállomás-függvények, például WSOSNameGet( ), WSOSVersionGet( ), WSLoginNameGet( ).

— Directory-függvények, például DrvDirMap( ), DirEffectiveRightsGet( ).

— Szemafor- és tranzakcióvédelmi funkciók, például TTSBegin( ), TTSEnd( ).

— A kézikönyvben mindenütt X-base-es pszeudokóddal írott mintaprogramrészletek.

Támogatott nyelvek: C/C++ nyelvjárások, Visual Basic 3.x, 4.x, Borland Delphi (itt a novlib.dll használatos, mivel a Delphi nem támogatja a statikus linkelést, valamint helyesen kezeli a Delphi string adattípusát is), Clipper Summer '87-en keresztül CA-Clipper 5.3-ig, CA-Visual Objects 1.0b stb.

Herczeg József

InContext:
Web Analyzer

Rendszer: Windows 3.1 vagy Win95
Kategória: Web utility
Dátum: 1996. május
Verzió: 1.0
Méret: 1140 K
URL: InContext Homepage

Egy HTML-oldal írásánál még nem nagy kunszt követni, hogy mire hivatkozik. Többnél — egy egész site esetén — ez már távolról sem olyan egyszerű. Mondjuk 30 oldal, 89 kereszthivatkozással és 217 ábrával eléggé próbára teszi az embert. És akkor arról még nem volt szó, hogy egy webes rendszergazdának nemcsak a maga által írott lapokat kell rendben tartania, hanem mások műveit is. Továbbá az ember néha kíváncsi lehet arra, hogy egy távoli site milyen szerkezetű.

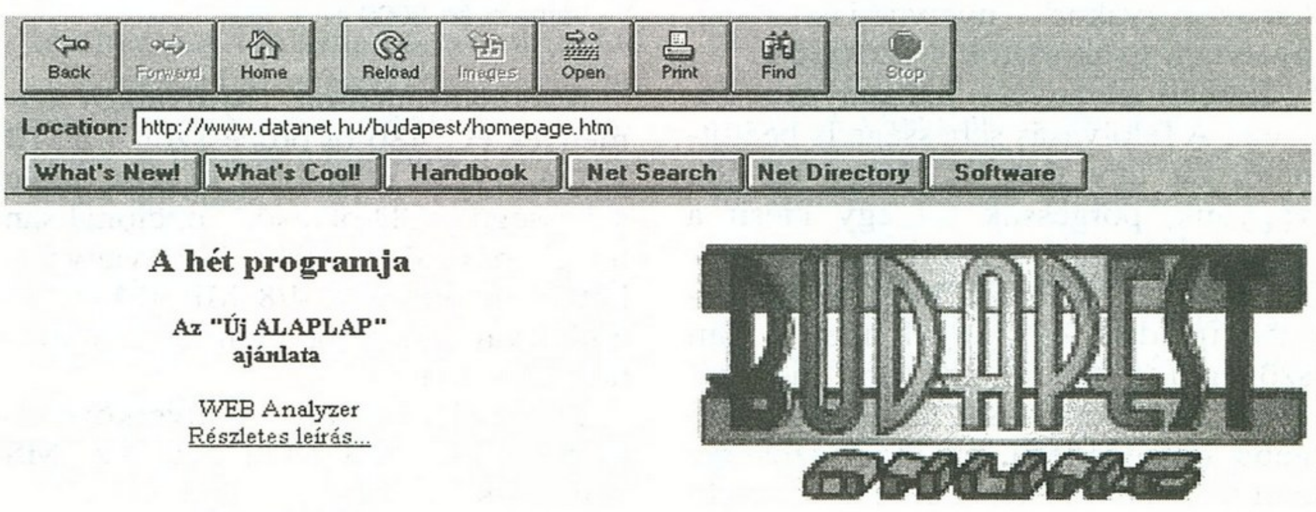
A fenti problémának tökéletes megoldása az Incontext Web Analyzere. A cég egyébként is jó nevű: IC Spider nevű web-szerkesztőjének újszerűsége miatt amúgy is megéri a figyelmet. A WA egy középpontként megadott HTML-oldalból kiindulva végigmegy az általa hivatkozott képeken, mail-címeken, HTML-oldalakon, majd ezt rekurzív módon folytatja, amíg valami miatt meg nem áll. Megállhat azért, mert a kiinduló oldal a mi winchesterünkön van, és ott már minden hivatkozást bejárt, vagy ha távoli gépről van szó, akkor valamelyik általunk megadott határfeltételbe ütközött. Ilyen feltétel lehet a hivatkozási mélység, illetve az, hogy külső hivatkozásokat (linkeket) már ne kövessen.

Az eredményt három ablakban láthatjuk. Jobb oldalt koncentrikus körökben a kiinduló oldal és annak hivatkozásai, majd a hivatkozások hivatkozásai vannak egészen addig, amíg lehet. Itt világosan érthető ikonok különböztetik meg az oldalt, a képet, illetve az egyéb elemeket. Például „meglepő" módon a mailto: címeket egy kis borítékikon reprezentálja. Bármelyikre a jobb oldali gombbal kattintva választhatunk, hogy nézegetni (browser) vagy szerkeszteni akarjuk (általunk megadott HTML-szerkesztővel), illetve a tulajdonságaira vagyunk kíváncsiak. A bal oldali felső ablak egy kiválasztott csomópont kapcsolatait mutatja, a jobb oldali gomb itt is hasonlóan működik. Látható, hogy egy kiválasztott oldal vagy egyéb elem mire hivatkozik, és mi hivatkozik rá. A nem létező kapcsolatok — törött linkek — jól láthatóan ki vannak emelve. Így egy elemzés után tudjuk, hogy aki nézegeti az oldalainkat, az hol fog szitkozódni, és mit kell tennünk, hogy ezt elkerüljük. Alul listaszerűen láthatjuk az oldalak és elemek felsorolását. Itt dátum, cím, hivatkozások száma, méret is látható. Külső, másik gép (site) esetén annak pontos URL-je is leolvasható ebből a listából. Talán mondani sem kell, az eddigiekből kiderül, hogy kívánságra az oldalak le is töltődnek gépünkre, tehát a G-A-S által nyújtott szolgáltatás, mintegy mellesleg, benne van ebben a programban.

Generálhatunk beszámolót is, és ez persze HTML-formátumú lesz, amit az általunk megadott böngészőben rögtön meg lehet nézni. Ebben minden követhető, amit egyébként a fent említett ablakokban esetleg azok zsúfoltsága miatt nem lehetett áttekinteni. Munkánkat természetesen elmenthetjük, és újra előhívhatjuk, nagy site-ok esetén félbeszakíthatjuk az elemzést, ami később onnan folytatható, ahol abbamaradt.

A program szellemes, ötletes, nagyon jól van megoldva, és kis helyen, kevés erőforrással nyújt mindent, amit ígér. Aki egy kicsit is komolyabban veszi a webet, az nem lehet meg nélküle. A demóváltozat ingyenes, letölthető a fentebbi címről, de még a teljes változat is megfizethető, 150 dollár alatt van az ára.

Horlai János



1996. február 1. — 1996. május 31.

Magyar CD-ROM diszkográfia

Bibliográfiai adatbázis

CIKK adatbázis

OMIKK Informatikai Igazgatóság;
Comet; Software AG; 1995—
Megjelenik negyedévente
1 példány ára 46 000,– Ft + áfa (F)
Éves előfizetés 80 000,– Ft + áfa (F)
* Kb. 120 000 cikk adatai mintegy 1500 szakfolyóiratból és egyéb dokumentumból. A lemezen lévő egyéb adatbázisok: K+F; Intézmények; Szakértők; Kutatási témák; Tankönyvek

MNB/CD

Magyar Nemzeti Bibliográfia.
Könyvek 1976–1995
Országos Széchényi Könyvtár;
Arcanum, 1996
ISSN 1218-2192
Éves előfizetése (a korábbi előfizetéstől és a hálózati használattól függően) 30 000 és 80 000 Ft között (F)
* Két évtized alatt megjelent kb. 125 000 magyar könyv bibliográfiai leírása

Teljes szövegű adatbázis

Népszabadság 1995

Népszabadság Kiadói Rt;
Hypermédia Systems, 1996
Ára 11 900,– Ft + áfa (k)

Jogszabálygyűjtemény, adattár

Adó '96

Devizaszabályok '96

Társadalombiztosítás '96

[Mindhárom fenti CD-hez azonos adatok:]
HVG Rt; HVG Press Kft.;
Folio Corporation, 1996
(HVG Infotár)
Ára 2950,– Ft [Egyenként]

CD-cégjegyzék

MTA SZTAKI; Company Data, 1995—
Megjelenik havonta
Regisztrációs díj 6500,– Ft + áfa
Előfizetési díj 2500,– Ft/hó + áfa (k)
* A Cégközlöny alapján kb. 250 000 magyar cég adatai

Jogikus

Közgazdasági és Jogi Kiadó;
Invenció Kft, 1996—
Éves előfizetés 39 000,– Ft + áfa
Darabonkénti ár 5000,– Ft + áfa
* Hatályos jogszabályok gyűjteménye 1993 júniustól

Samsung

Irodatechnika termékkatalógus '96
Samsung Electronics Magyar Rt, 1996
Ingyenes, kapható a cégnél és partnereinél

HMS 1996

Hypermedia Systems, 1996
Cégbemutató
Ingyenes

Multimédia adatbázis

A magyar koronázási jelvények

Magyar Nemzeti Múzeum;
Cognitech, 1996
(Magyar Nemzeti Múzeum, CD Könyvtár, 1.)
Ára 3990,– Ft (k)

Fegyvertár

Válogatás a Magyar Nemzeti Múzeum fegyvergyűjteményéből. Magyar Nemzeti Múzeum; Info2000, 1996
(Magyar Nemzeti Múzeum, CD Könyvtár, 2.)
Ára 3990,– Ft (k)

Városképek — 1990

Magyarországi városképek a századfordulón,
Magyar Nemzeti Múzeum;
Info2000, 1996
(Magyar Nemzeti Múzeum, CD Könyvtár, 3.)
Ára 3990,– Ft (k)

Emese álma

A magyar őstörténet és az államszervezés kora (A kezdetektől 1038-ig)
Enciklopédia Humana Egyesület, 1996
Ára 6000,– Ft + áfa (F)

Budapest

CD-atlasz
Cartographia; Vision-X, 1996
Ára 8700,– Ft (KK)

Képtárak

Kirándulás a képzőművészet világába
Minor, 1996
Ára 5200,– Ft (KK)

Révai nagy lexikona

CyberStone, 1996
4 db CD ára 19 900,– Ft (F)

Úton

'96-os KRESZ CD
CyberStone, 1996
Ára 3990,– Ft (F)

Autósiskola

Multimédia oktató és vizsgáztató program
TalmaMédia, 1996
Ára 6800,– Ft (KK)

CD-ROM+

Kossuth, 1996
Ára 1300,– Ft (KK)
* Az önállóan is kapható CD egyben „A multimédia alapjai" c. könyv melléklete

PC-ROM

Multimédia PC Enciklopédia
Foxtrend, 1996
Ára 7790,– Ft (KK)

Nyelvoktató program

ClipDIC Deutsch 1

Komplex nyelvoktató CD közép- és felsőfokú nyelvvizsgára készülőknek
Profi-Média, 1996
Ára 6200,– Ft + áfa (F)

ClipDIC Business & Finance

English Tutorial
Profi-Média, 1996
Ára 7200,– Ft + áfa (F)

Mesevilág 1.

A három medve és a kisleány & más mesék
Kossuth, 1996
(Story World 1)
Ára 5500,– Ft (KK)

Mesevilág 2.

Piroska és a farkas & más mesék
Kossuth, 1996
(Story World 2)
Ára 5500,– Ft (KK)

Nyelvmester

Nyelvoktató CD: Haladó angol
Vision-X, 1996
Ára 6990,–Ft (KK)

PC Suli

Multimédiás újrakezdő és középhaladó angolnyelv-oktató CD-ROM
Zombie Soft, 1996
Ára kb. 8000,– Ft + áfa

CD-ROM-on rendszeresen megjelenő periodikum

Other Side

Multimédia PC Magazin
Other Side Magazin, 1996—
Megjelenik kéthavonta
Ára 990,– Ft/szám (F)

Video CD

Fekete doboz

Fekete Doboz Alapítvány;
Com-Ser, 1996
Ára 2000 Ft + áfa (k)

Egyéb CD-ROM

Magyar fürdőkalauz

Magyar Fürdőszövetség;
Magyar Almanach Kiadó; Helion, 1996
Ára 2500,– Ft (áfával együtt)

Virtual Gallery I., II. és II/2.

ArchiCAD & TopCAD
Application & Sample Source CD-ROM
Graphisoft, 1993—1995
Macintosh gépen fut
Ára 2400,– Ft + áfa (II/2. verzió: k)

ArchiCAD Gallery 2.0

Graphisoft, 1995
A Virtual Gallery PC-s változata
Ára 2400,– Ft + áfa (k)

People and more

Graphisoft, 1996
(Graphisoft Collection, vol. 1)
Macintosh gépen fut
Ára 19 500,– Ft + áfa (k)

A zárójeles rövidítések a megrendelés módjára utalnak:
(k) — A kiadónál
(F) — A fejlesztőnél
(KK) — A Kossuth Kiadónál
A listával kapcsolatos észrevételeket és kiegészítéseket köszönettel fogadja a lista összeállítója:

Tószegi Zsuzsanna

INFORMÁCIÓKÉRÉS: 0712 ▲

INFORMÁCIÓKÉRÉS: 0727 ▲

INFORMÁCIÓKÉRÉS: 0707 ▲

Kliens/szerver alkalmazásfejlesztés II.

A feladathoz igazodva

A munkában, melynek eszközeiként választhatjuk az elmúlt havi cikkben ismertetett elemeket, kulcsszerepe van az „adatrendezésnek". Az akkor indokolt ajánlást (1996/6, 28. old.) ebben a szakmai megvilágításban részletezzük.

A Powersoft Portfolio vizsgálatának második részét kezdjük mindjárt a végén, azzal az adatbáziskezelővel, amelyre terveinket alapozhatjuk. Egyébként ugyanennek egyfelhasználós, Windows alatti változatát, mint fejlesztési környezetet, a PowerBuilder Desktop programmal együtt is megkapjuk. Ez tartalmazza a fejlesztőeszköz demóalkalmazását is.

Watcom SQL — a motor

A háromfelhasználós Watcom SQL a munkacsoportos, az egyfelhasználós, illetve a mobil felhasználók számára készülő alkalmazások fejlesztésének hatékony eszköze, amely teljes vállalatot átfogó hálózatig rugalmasan továbbfejleszthető. Minimális adminisztrációs tevékenységet igényel, könnyen telepíthető és karbantartható. A többplatformosság révén van átjárás a Windows, Windows 95, Windows NT, OS/2, NetWare és DOS környezetek között.

Figyelemre méltó a Watcom SQL Server kis memóriaigénye (kevesebb, mint 1 Mbájt), ami a hordozható gépeken futó alkalmazások szempontjából sem közömbös. A Server kapcsolódásonként is csupán 2 Kbájtot igényel. A Server ún. költségalapú lekérdezésoptimalizáló eljárása futás közben választja ki a legmegfelelőbb módot az információ elérésére.

Fejlett adatbázis-funkciók segítik az eljárások, triggerek használatát, a kaszkád update-et és törlést; a kétirányú, görgethető és update-elhető kurzorokat. A teljes tranzakciófolyamatot ellenőrzőpontokkal és tranzakciónaplózással támogatja.

Egyszerűbb feladatok megoldásakor gyakran egybemosódik az adatstruktúrák logikai és fizikai tervezése. Ugyanakkor nagyobb szerepet kap már létező adatmodellek visszafejtése, korrigálása, áthelyezése. Tekintve, hogy a fejlesztés kis létszámú, különös szerepe van a létrehozott modellek megfelelő dokumentálásának is. Ezekhez nyújt hatékony segítséget a StarDesignor, amelynek „felnőtt" testvére az S-Designor Enterprise. (Ez utóbbi jóval szélesebb szolgáltatási körének ismertetésére itt nem térhetünk ki, de jó, ha szem előtt tartjuk: ha megoldandó feladataink meghaladnak egy bizonyos bonyolultsági szintet, célszerű lesz tájékozódni ebben az irányban!)

A StarDesignor 1.0 olyan (grafikus) adatbázistervező eszköz, amelyben a táblák, oszlopok és a köztük lévő kapcsolatok megrajzolásával, az elemek tulajdonságainak megadásával (vagyis a tervezés lépéseivel) egyidejűleg a kialakított adatbázis dokumentálását, illetve fizikai létrehozását is meg tudjuk valósítani.

A program több mint 50 adatbázisplatformot támogat. A Watcom SQL Server adatbázissal teljesen integrált egységet képez, míg a többi esetben ODBC meghajtókat használ. Adatbázis-visszafejtő technológiája lehetővé teszi a már létező adatbázis átvilágítását, így grafikusan megtekinthető és módosítható annak szerkezete.

Prezentációs színvonalú jelentések is generálhatók, és lehetséges az adatbázis áttelepítése akár másik adatbáziskezelő rendszer alá.

A tárolt eljárások és triggerek támogatásán kívül az üzleti és érvényesítési szabályok definiálási lehetősége még hatékonyabbá teszi az alkalmazást, előnyt biztosít az alkalmazástervezésben.

Az adatbázis-szerkezet létrehozása után a PowerBuilder Desktop sokrétű, egyfelhasználós fejlesztői környezete áll rendelkezésünkre az alkalmazás létrehozásához. (Ennek a programnak is léteznek nagyobb „testvérei": a PB Team-OBDC támogatja a csoportos fejlesztéseket, míg a PB Enterprise ezenfelül natív meghajtókat is biztosít alegelterjedtebbadatbáziskezelőkhöz. Ezen a termékcsaládon figyelhetjük meg leginkább a méretezhetőség célszerűségét, amit az árak 1:15 arányú különbsége is éreztet!)

A Windows környezetben futó program lehetővé teszi, hogy a rámutatás, kattintás, vonszolás kényelmes módszerével fejlesszünk. Az elkészült alkalmazást egy menetben fordíthatjuk le kész programmá, amely jogdíj és költséges futtató rendszer nélkül közvetlenül alkalmazásba állítható.

Az alkalmazásfejlesztés korszerű eszközeit biztosítja a program. Segítségével gyorsan, bonyolult programozás nélkül fejleszthető komplex, minden funkcióval ellátott alkalmazás. Az objektumorientált technológia tartalmazza a többszintű öröklést, a logikai beágyazást, a polimorfizmust, és az osztálykönyvtárak támogatását. A PowerScript programozási nyelvben több száz beépített függvény használható, felhasználói függvények írhatók, és elérhetők a C-ben vagy más nyelven írt funkciók is. A rendszer tartalmazza a fordítót, valamint a teljes körű nyomkövetés lehetőségét is.

Az építő

A desktop integráció megkönnyítésére a rendszer támogatja az OLE 2.0 csatolásokat, a dinamikus adatcserét (DDE), a DLL-eket és a Visual Basic kontrollokat (VBX). A gyorstipp lehetőséggel is ellátott konfigurálható eszközsorok a munka hatékonyságát növelik.

Az eszközsorok testre szabhatók, mozgathatók, és gyorsan konfigurálhatók. Kiterjesztett attribútumtámogatás teszi lehetővé a megjelenési és érvényesítési formátumok, szerkesztőmaszkok és oszlop-kezdőértékek kezelését.

A PowerBuilder intelligens DataWindow objektumával az adatbázis SQL programozás nélkül is manipulálható. A DataWindow értelmezi minden adatbázis egyedi tulajdonságait: egyszerűen csak ki kell választani az adatforrást és a megjelenítési stílust, és a DataWindow közvetlenül az adatbázissal bonyolítja le a párbeszédet.

A rendszer ODBC-n keresztül kliens elérést biztosít többek között Dbase, Clipper, FoxPro, Paradox, NetWare SQL, Btrieve, ASCII text, Excel adatokhoz.

A pipeline a PowerBuilder adatmozgató és másoló eszköze: táblákat, táblák adatait és a tárolással kapcsolatos in-

formációkat másolja egyik adatbázisból a másikba, még ha az adatok különböző helyen vagy másik adatbáziskezelő rendszerben vannak is. Egy már létező adatbázis-szerkezet StarDesignorral való honosítása után a PowerBuilder Desktop pipeline technológiájával az adat a forráshelyről áttölthető a Watcom SQL Serverbe, vagy nem hálózatba kötött Access, Dbase és Paradox adatbázisokba.

Könnyen készíthetők a rendszerben lekérdezések, riportok és üzleti grafikák a grafikus lekérdezésszerkesztővel, illetve a többtáblás QuickSelect eszközzel. A lekérdezések objektumként elmenthetők, majd adatforrásként használhatók különböző (szabad formátumú, tabulált, kereszttáblás stb.) riportok számára. Készíthető 2 és 3 dimenziós tortaszelet, sor-, oszlop-, vonal- és pontdiagram, egyéb stílusú grafikon is.

Bővíthetőség, méretezhetőség

A fejlesztés hatékonysága tovább növelhető a Powersoft egyéb termékeinek hozzáadásával. Az Advanced Developer Toolkit újra használható objektumok, eszközök, segédletek gazdag gyűjteménye. A népszerű Funcky for PowerBuilder a PowerScript nyelv kibővítése 500-nál több függvénnyel. A PowerBuilder Library for Lotus Notes a Notes adatbázisokhoz való kapcsolódást teszi lehetővé. A ComponentPack a táblázatok és helyesírás-ellenőrző képességek beépítését segíti. A Translation Assistant a nemzetközi terjeszthetőséghez szükséges többnyelvű alkalmazások fejlesztésének támogatására készült.

Ha a fejlesztői csoport vagy a felhasználói bázis növekedése indokolttá teszi, a Powersoft Portfolio lehetőségei egyszerűen növelhetők a vállalati szintű modulokra bővítéssel. A már említett S-Designor Enterprise és PowerBuilder Enterprise komplex környezetet nyújt a munkacsoportos tervezéshez és fejlesztéshez. A Watcom SQL Advanced Network Serverek megadják a lehetőséget a háromfelhasználós környezetről a teljes vállalatot átfogó méretű rendszerre való áttérésre.

Bátorfi Péter—Szabó György

Megint egy új fogalom: WYNIWYG

Internet web-oldalak nyomtatása

Mi lehet az oka annak, hogy a számítástechnikában bizonyos egységesítési törekvéseket a felhasználók és a gyártók egyaránt szívesen, másokat viszont igencsak berzenkedve fogadnak. Érzelmileg is alaposan megosztja például a PC-s világot a Windows platformkényszere, ugyanakkor nem nagyon lehet az „egyirányú utca" miatti panaszkodást hallani a Postscript leírónyelv vagy a Norton Commander „diktatúrája" esetében. A kérdésre adandó mélyebb válasszal az alábbi beszámolóban is adósak maradunk, de ha olvasóinknak kedvük támad ezzel kapcsolatos véleményüket kifejteni, szívesen közreadjuk azokat. Mi most az egység és a sokszínűség felé egyszerre haladó világból egyetlen aktuális darabkát fogunk csak kiragadni.

A számítástechnika történetében már többször előfordult, hogy a Hewlett-Packard a többiek előtt járt, és kirukkolt olyasmivel, ami utána szabvánnyá, másolási mintává, fejlesztési trendek meghatározójává vált. A kutatásra és fejlesztésre koncentráló „mérnökcég" nem a raffinált marketingnek, nem is valami titkos ipari összefonódásnak vagy egyéb machinációknak köszönhette sikerét, hanem termékei meggyőző minőségének — és feltehetően annak, hogy nem a jelenben, nem is a holnapban, hanem legtöbbször a holnaputánban gondolkodott.

A fenti asszociációkat a Hewlett-Packard bergamói gyárában látottak és hallottak ébresztették fel a meghívott újságírókban. Maga a gyár, annak szervezettsége, felszereltsége, tisztasága olyan, mintha az egész valahol Kaliforniában lenne. A téma nem is ebben rejlik, hanem a bemutatóhoz kapcsolódó előadásokban, azon belül is leginkább a hálózati nyomtatásban.

A dolog ott kezdődik, hogy bár számos szép „szürke elmélet" született a papír nélküli irodáról, az élet „zöld fája" egyre inkább úgy fest, hogy még a távoli jövőben is igen előkelő szerepe lesz a nyomtatásnak, a papírnak. Persze nem egészen abba a formában, mint jelenleg. A szélessávú adatcsatornákon online elérhető újságokból és folyóiratokból a magunk érdeklődése szerint szelektálva otthon vagy az irodában kinyomtatott oldalak persze még a távoli jövőképhez tartoznak, de már „válámi van". Ez a bizonyos valami az internetes és intranetes technológia, azon belül is a grafikailag kidolgozott, képekkel illusztrált web-lapok gyors terjedése. És amit a hálózaton látunk, azt esetenként jó lenne a nyomtatóra is kiküldeni, minél jobban megközelítve a képernyőn látható eredetit. Megjelent tehát a WYNIWIG fogalma (What You Net Is What You Get), vagyis amit a hálózatból be tudsz hívni, azt ki is tudod nyomtatni.

A baj csak az, ha a piaci versenytársak hasonló technológiáiból hiányzik az egység, ha mindegyik a sajátját akarja futtatni, amely nem teljesen kompatibilis másokéval. A gondolatmenet itt visszakanyarodik a bevezetőhöz: ha az uniformizálási törekvést a szakma — és a felhasználók szélesebb rétege — az Internet felületkezelésében rivalizálóktól nem fogadja el, talán a HP-tól mint kívülállótól és nyomtatási ügyekben igazán autentikus forrástól igen. Az Infoworld írta 1995. december 11-i számában: „A Microsoft és a Netscape HTML-verziói eltérő módon támogatják a nyomtatást. A HP viszont elég nagy ahhoz, hogy meghatározza, mi legyen az egységes szabvány."

Azt nem nagyon kell bizonygatni, milyen problémák fakadnának az egységes szabványok hiányából, és az sem válik az emberi faj dicsőségére, ha a normákat nem a rendelkezésre álló legjobb megoldásra építik. (Bár ilyen modellek is működőképesnek bizonyulhatnak, nem is akármilyen elterjedtséggel, és ki-ki vérmérséklete szerint veszi az ilyen tényeket is tudomásul.)

A HP a feljesztési célok meghatározásánál elemezte, hogy melyek a felhasználók leggyakoribb panaszai a hálózatról történő nyomtatáskor: lassú a nyomtatás, túl sok lépésből áll, a grafikák gyenge minőségűek, bizonyos részek lemaradnak vagy hiányosak, az oldalrészletek különálló kinyomtatása megoldatlan, „szemetelő" karakterek jelennek meg az oldalon, rosszul vagy csúnyán vannak kinyomtatva a betűk, a nyomtatás lelóg a papírról stb.

Mindenekelőtt ki kellett fejleszteni a HP nyomtatási lapleíró nyelvcsaládjának az Internet web-technológiáját kezelni tudó legújabb tagját, a PCL 6-ost. Kooperáció nélkül a problémák megoldásának sokkal kisebb az esélye. A HP azt az utat választotta, hogy 1995 decemberében közös munkába kezdett a Microsoft és a Netscape szakembereivel, és 1996 márciusában már be is jelentették a W3C közös fejlesztésű stíluslapot. A web-oldalak univerzális kinyomtathatósága felé vezető úton a szakma tehát elindult.

Más vonatkozásban a HP új eszközök sorával állt elő. A nyomtatási sebességről, a MOPY-technika terjedéséről (lásd erről legutóbbi számunkat) a HP LaserJet 5Si és az 5Si MX nyomtató gondoskodik, percenkénti 24 oldalas (A/4-es) teljesítménnyel. A papíron lévő dokumentumok gyors digitalizálását és a hálózaton való továbbítását végzi el a ScanJet 4Si szkenner. A képek minőségének javítását szolgálja a színes printerekhez kifejlesztett HP Image REt 1200-as technológia, amely például a Color LaserJet 5 esetében a 300 dpi-s berendezésen 1200 dpi-vel egyenértékű nyomtatási képet szolgáltat, több millió színárnyalat alkalmazásával.

Az internetes kihívásra a HP elég gyorsan reagált, mert valószínűleg nem akkor kezdett el a megoldáson dolgozni, amikor a hálózaton az első HTML-oldalak megjelentek.

Faklen Pál

Személyi hír

Május 24-én a Microsoft Magyarország élén Gönczi Andrást a cég eddigi kereskedelmi igazgatója, Reisz Attila váltotta fel. Gönczi András a redmondi központban a Microsoft Network online szolgáltatások nemzetközi marketingvezetőjeként dolgozik tovább.

AT&T → Lucent Technologies

Egy patinás név — az AT&T — Magyarországon lényegében eltűnik a szakmai palettáról. Az amerikai piacon ott marad a tisztán távközlési szolgáltatási profilt továbbvivő AT&T, a három részre darabolt céghez kapcsolódó egyéb tevékenységek viszont más nevek alatt folytatódnak. A berendezések gyártása és forgalmazása révén nálunk megismert AT&T név mögötti tevékenységeket az AT&T Magyarország Kft teljes jogú utódvállalataként a Lucent Technologies Magyarország Kft. folytatja, amely a Lucent Technologies Business Communications Systems üzletág közvetlen képviselője. Ez a vállalat épületen belüli és épületek közötti integrált multimédia-hálózatok (IMX) megvalósításával foglalkozik, cégek és intézmények számára. Ugyancsak a Lucent Technologies Magyarországi Kft-n belül, de szervezetileg a Lucent Technologies Network Systems üzleti egységéhez kapcsolódva működik a Privát Hálózatok Üzletág, amely a nevében foglalt szolgáltatásokra szakosodik, és indirekt csatornákon keresztül értékesít Magyarországon.

Kicsit bonyolult ugyan, de idővel majd megtanuljuk. Erre mondhatja a pesti ember: mindegy, hogy hogy hívják, a lényeg, hogy AT&T legyen...

Packard Bell + NEC!

A korábbi hírek alapján már számítani lehetett a június 4-én bejelentett tényre: a NEC a személyi számítógépekkel kapcsolatos működését Japánon kívül szorosan a Packard Bellhez köti. A közös vállalat a Packard Bell NEC nevet viseli. Az előzményekhez tartozik, hogy a Packard Bell áprilisban átvette a Zenith Data Systemset a Bull csoporttól. A Packard Bell (NEC) tehát amerikai hódításait az egész világra ki kívánja terjeszteni, s deklaráltan világelsőségre törekszik. A Packard Bell Amerika kedvenc számítógép-márkája lett, a fejlett, erősen integrált és jól kezelhető multimédia-technológiát ők juttatták el a legtöbb emberhez. A mostani tranzakció annak lehetőségét teremti meg, hogy a kombinált infrastruktúra, az egyesített K+F révén a Packard Bell NEC a családi, intézményi és vállalati piacok fő kiszolgálójává váljon.

Itthoni vonatkozású hír a témához kapcsolódóan, hogy tovább gazdagodik a Packard Bell gépekkel együtt szállított, azokra előre installált programok választéka: DOS vagy Windows, Accent Express és egyéb hasznos szoftverek mellett június 1-jétől a Hunix Kft népszerű VirusBuster programcsomagja is felkerül a gépekre. A programcsomag nem demó, hanem a gépvásárláskor érvényes „éles" változat (a jelenlegi például már tartalmazza a makróvírusok ellenszerét is). A felhasználó később már az alkalmazási tapasztalatok birtokában döntheti el, hogy az ajándékba kapott vírusellenes programcsomag további frissítéseire igényt tart-e.

Armada a Dunán

Június 19-én stílszerűen egy állóhajó fedélzetén jelentették be a Compaq legújabb hordozható számítógépének, a Compaq Armadának a forgalomba kerülését. A szlogen szerint az év Compaq-meglepetése, az Armada az a notebook, amely „nem ismer kompromisszumokat". A cég által kiadott sajtóanyagból idézve: „Eddig, ha valaki komoly technikai tudású mobil-computert vásárolt, számítania kellett rá, hogy feltöltés nélkül csak rövid ideig működik. A jó felbontású képernyő ugyanilyen hátránnyal járt, ha pedig valaki kiegészítőket is vásárolt notebookjához, hamarosan megizmosodott, annak meglehetős súlya miatt."

„Az Armada ... *legoszerűen* építhető rendszere lehetővé teszi, hogy a gép egyszerre legyen tudásban nagy, formáját és súlyát tekintve kicsi. Konstruktőrei úgy tervezték meg, hogy illeszthető legyen tulajdonosának asztali számítógépéhez is — így kapcsolódhasson akár a teljes munkahelyi hálózatához is —, ugyanakkor egy mozdulattal kivehető legyen onnan, az összes információval együtt."

Ennyi jó dolog felsorolását követően kíváncsian várjuk, miként vizsgázik az Armada a gyakorlatban, s tekintettel arra, hogy mi egy darabig még nem engedhetjük meg magunknak ezt a fajta „hajózást", várjuk az Armada-tulajdonosok visszajelzéseit.

JDE a Rolitronnál

„Az érték a termelésben keletkezik" — olvasható valahol egy Rolitron-hirdetés szlogenje. Hogy ennek az értéknek a tisztelete mennyire hangsúlyos a Rolitron stratégiájában, azt mi sem bizonyítja jobban, mint a vállalatirányítási rendszerekre szakosodott önálló divízió a cégen belül (ERP = Enterprise Resource Planning). A szakaszos termelés irányítására már eddig is kínált Symix és a folytonos termelésre alkalmas Prism mellett harmadikként az ugyancsak szakaszos termelést kiszolgáló J.D. Edwards rendszer került fel a palettára. Fontosságát tekintve ez utóbbi akár a másik kettő elé is kerülhet. A szerződés értelmében a világ legfontosabb öt ERP-rendszere között számontartott J.D. Edwards üzleti partnere a Rolitron Informatikai Rt.

Maga a JDE rendszer nagyon széles felhasználói területet képes lefedni, a gyártástól a disztribúció/logisztika és a pénzügy/számvitel területén át a különböző szolgáltató tevékenységekig. (Funkcionalitását tekintve lényegében az SAP-vel van azonos szinten.) A programrendszer negyedik generációs adatbáziskezelőt, fejlett CASE-eszközöket és sokrétű vállalati szolgáltatást nyújt. Az objektum-orientált eszközökkel fejlesztett változat piaci bevezetése most kezdődött az USA-ban, s a gyakorlati próbák után 1997 közepétől a magyarországi felhasználók rendelkezésére is áll majd.

FLaG Party — nyitott kapukkal

Májusi számunkban hírül adtuk, hogy Budapesten FLaG néven nemzetközi számítástechnikai találkozót szervez a Controlled Dreams demócsapat. A rendezvény az akkor meghirdetettekkel szemben nem három, hanem négy napig tart majd, 1996. július 19-től 22-ig, a III. kerületi Bárczi Géza Közösségi Iskola épületében (Bárczi Géza utca 2.) A szervezők a partyra szeretettel várnak mindenkit, akik a C64, az Amiga vagy a PC megszállottjai, de a kapuk nyitva állnak az egyszerű érdeklődők és a szerepjátékosok előtt is.

Kétüléses

Ha a „tandem" szóról valakinek csak a kétüléses kerékpár jut az eszébe, abban egy kicsit a COCOM előírásai is ludasak voltak. Az 1974 óta működő Tandem Computers Inc. ugyanis a szuperszámítógépek csoportjába sorolva sokáig el volt zárva a kelet-európai piacoktól. Olyan rendszereket szállított, amelyekkel szemben igen szigorúak voltak a megbízhatósági követelmények. Ma is a Tandem megoldásain alapul a világ hitelkártya-tranzakcióinak 66%-a, 200 egészségügyi és biztosítási szervezet készpénzelosztó rendszereinek 75%-a, a 32 legnagyobb távközlési vállalat hívásirányítási rendszereinek 50%-a. A Tandem számos szakterületen (online tranzakciófeldolgozás, párhuzamos adatbáziskezelés stb.) úttörő szerepet játszik. Nem csoda tehát, hogy a Microsoft komolyan érdeklődni kezdett iránta, és 1996. május 7-én stratégiai szövetségre lépett vele. A kritikus alkalmazásokban, a non-stop technológiákban, az adatkezelésben elérhető előnyökön kívül ez a kapcsolat azt is jelzi, hogy a nagy teljesítményű operációs rendszerek közül a Tandem is a Windows NT-t részesíti előnyben, feltehetően a Unix rovására.

APRÓHIRDETÉSI MEGRENDELŐLAP

Kérem, hogy az Új Alaplap következő számának
Mikrobazár rovatában közöljék az alábbi szövegű apróhirdetést:

....

....

....

....

....

(Maximális terjedelem: 300 betűhely)

Alaplap Posta

MEGRENDELÉS

Az Új Alaplap 1996/7. számában a 25-26. oldalon ismertetett **3 szoftver közül postai utánvétellel** megrendelem az alábbiakat:

Név: ....

(Cég:) ....

Cím: ....

Helység: ....

Irányítószám: ....

A szoftverek árát a küldemény átvételekor
a kézbesítési díjjal együtt kifizetem.

Dátum:

/aláírás/

Előfizetés az Új Alaplapra

Az 1996/.......... számtól kezdődően előfizetem az Új Alaplap c. havi számítástechnikai folyóiratot

...... példányban, ❑ 1 évre, ❑ 1/2 évre.

Az éves előfizetési díj 3564,– forint.

Az előfizetési díj kiegyenlítéséhez:

❑ **Számlát** kérek (banki átutalással fizetek).

❑ **Átutalási postautalványt** kérek.

Név: ....

(Cég:) ....

Cím: ....

Helység: ....

Irányítószám: ....

Dátum: ....

- - - - - - - - - - - - - - - - -

/aláírás/

INFORMÁCIÓKÉRÉS

Az Új Alaplap 1996. júliusi számának hirdetéseihez

Kérem, hogy az itt általam **BEKARIKÁZOTT KÓDSZÁMÚ** hirdetésekkel kapcsolatban küldjenek részemre bővebb tájékoztatást.

Beküldhető: 1996. július 31-ig

0601	0612	0623
0602	0613	0624
0603	0614	0625
0604	0615	0626
0605	0616	0627
0606	0617	0628
0607	0618	0629
0608	0619	0630
0609	0620	0631
0610	0621	0632
0611	0622	0633

FELADÓ **Feladáskor kérjük bérmentesíteni!**

A) Egyéni érdeklődő:

Név: ....

Cím: ....

Helység: ....

Irányítószám: ....

B) Vállalati érdeklődő:

Cég: ....

Ügyintéző: ....

Cím: ....

Helység: ....

Irányítószám: ....

Telefon/Fax: ....

Bélyeg helye

Új Alaplap
szerkesztősége
I., Márvány u. 17.
Pf. 571
Budapest 1539

Belföldön díjmentesen feladható

Új Alaplap
szerkesztősége
I., Márvány u. 17.
Pf. 571
Budapest 1539

Minden PC-hez
kell egy jó alaplap!
És egy Új Alaplap!

Belföldön díjmentesen feladható

Új Alaplap
szerkesztősége
I., Márvány u. 17.
Pf. 571
Budapest 1539

FELADÓ: **Feladáskor kérjük bérmentesíteni!**

Név: ....

Cím: ....

Helység: ....

Irányítószám: ....

Telefon: ....

❑ A hirdetés egyéni és egyedi jellegű, ezért kérem ingyenes megjelentetését. Kijelentem, hogy annak tartalma nem sérti senki szerzői jogát.

❑ A hirdetés kereskedelmi célt szolgál. Mellékelem a soronként (60 karakterenként) 300 forintnak megfelelő összeg átutalásáról az igazoló szelvény másolatát.
A címzett: Új Alaplap, 1539 Budapest, Pf. 571, illetve átutalásnál az OTP 11701004-20171649 számlaszám.

Bélyeg helye

Új Alaplap
szerkesztősége
I., Márvány u. 17.
Pf. 571
Budapest 1539

Az Új Alaplap lemezmellékletének tartalma — 1996 / JÚLIUS

3Com: Ethernet/ATM kapcsoló

Az amerikai „Computer Reseller News" c. szaklap egyik legutóbbi számában közölte a Strategic Consulting, Inc. különböző gyártók Ethernet/ATM kapcsolóit összehasonlító vizsgálatának eredményeit. A szempontok között szerepelt az átbocsátás, késleltetés, csomagvesztési arány (PLR), túlterheléstűrés és a hálózatkezelési szabványoknak való megfelelés.

Az ár/teljesítmény viszonyt tekintve a 3Com „újdonsült" LinkSwitch 2700 terméke került az élre. Az átviteli sebességek, az átbocsátás és a csomagvesztési arány tesztjében a LinkSwitch 2700 bizonyította a teljes átviteli sebesség mellett azt is, hogy minden csomagméret mellett 100%-os az átbocsátása. Ennek következtében nem volt csomagvesztés. Az OC—3c—ATM összeköttetésen mért Ethernet—Ethernet forgalom tesztjei is 100%-os átbocsátást mutattak. A túlterheléstűrés vizsgálatakor a LinkSwitch 2700 négy, öt és hat bemeneti jelfolyam (stream) mellett is azonos szintű, nagy teljesítményt mutatott egyetlen kimeneti porton. Ez független volt a kerethossztól. Ezt a teljesítményt egy a 3Com által kifejlesztett, „ZipChip" névre keresztelt ASIC lapka nyújtja. Az Ethernet/ATM között kapcsolt hálózat esetén szükség van egy csekély késleltetésre, amit a tesztek ki is mutattak. A vizsgált kapcsolók közül egyedül a LinkSwitch 2700 kínált választást a „store and forward" és a „cut through" üzemmód között.

AT&T—SGI: stratégiai szövetség

Az AT&T és a Silicon Graphics (SGI) június elején jelentette be, hogy stratégiai szövetségre lép egymással WWW Internet és Intranet rendszerintegráció szolgáltatására Európában, a Közel-Keleten és Afrikában. A megegyezés eredményeként az AT&T egyben a teljes SGI-termékskála viszonteladójaként fog működni. A hír jelentőségét növeli, hogy a távközlési óriás először lép ilyen szoros szövetségre egy vezető számítógéprendszer-gyártóval. Hírek szerint az AT&T teljeskörű tanácsadást nyújt majd, és biztosítja az integrált SGI—AT&T megoldások üzembehelyezését. A szövetség működésének támogatására Párizsban szakértői centrumot állítanak fel.

HP—Netscape: Internet-együttes

A Hewlett-Packard és a Netscape Communications üzleti célú szövetségre lépett, amelynek célja a vállalati internetes és intranetes hozzáférési piacok terén mutatkozó növekvő igények jobb, gyorsabb kielégítése. A partnerkapcsolat — amely „egyesít" egy főleg a hardvertermékeiről ismert vezető céget egy olyan szoftverforgalmazóval, amelynek neve már majdnem az Internet szinonimájává vált — széles területeket ölel fel, beleértve technológiai együttműködést, közös termékfejlesztést, maerketinget, eladási procedúrát, támogatási, konzultációs és oktatási tevékenységet. A létrejött HP—Netscape szövetség révén több projekt megvalósítására is lehetőség nyílik. Így kiterjesztett alkalmazási programinterfész eszközök (API-k) létrehozására; Internet-menedzselő megoldások előkészítésére, mind Unix, mind Windows NT környezetben, felhasználva a Netscape programszerver termékeit és a HP OpenView szinte már szabványnak tekinthető rendszerfelügyelő szoftverét; a HP OpenMail és a Netscape Mail Server elektronikus levelező rendszerek közötti együttműködésre, vállalati e-mail alkalmazásokban; a HTML nyelv nyomtatás-orientált kiterjesztéseire és a Netscape Web-böngészőjének a HP Laserjet nyomtatóit is felhasználni tudó, testreszabott változatának kidolgozására lesz lehetőség.

Lotus: megállíthatatlan Notes

Olyan Notes-verziót dob a tömegfelhasználói piacra a Lotus, amelynek funkciói eddig csak a nagy szolgáltatók, a nyilvános hálózatok részére voltak elérhetők. A két legfőbb „újítás": a megbízhatóságot jelentősen növelő cluster-funkció (klaszterezés, fürtözés) támogatása, illetve a groupware programcsomag teljesítményének a növelését célzó párhuzamos replikáció megvalósítása. A cluster-funkció több szerver olyan összekapcsolását jelenti, amelynek révén együttműködhetnek alkalmazások hatékony feldolgozásában. Jelenleg a Notes-felhasználók ún. proxy szerverek használata révén „kerülhetik meg" a problémát, amikor egy Notes szerverben hiba lép fel. Ilyenkor másik Notes szerverre „kapcsolnak át". A proxyk tipikusan azt igénylik, hogy a felhasználók a szerverek között manuálisan kapcsoljanak át, míg a klaszterezés automatikusan „átirányítja" a felhasználókat. Hírek szerint a Notes további új funkciókkal is rendelkezik majd, így például tükrözött e-mail állományok támogatása szerepel a tervekben.

A Java az NT-ben is elérhető lesz

A Microsoft bejelentette, hogy a Windows és a Windows NT operációs rendszer egyaránt támogatni fogja a Sun Microsystems Java programnyelvét. A Microsoft-féle Java-implementációt — amely lehetővé teszi a Windows platform speciális szolgáltatásainak (például hang, grafika stb.) használatát, a Java referencia-implementációjának tekinti a Sun ezen a platformon. A Java megjelenésével a Windows környezetben az válik lehetővé, hogy a Java programozásban kihasználják a Microsoft ActiveX objektumtechnológiájának előnyeit, például azáltal, hogy a Weben rendelkezésre álló ActiveX objektumokat építenek programjaikba, megtakarítva ezzel a szoftverfejlesztési időt. Hírek szerint a Microsoft a Java implementációt tovább javítja, kiterjeszti egy nagy teljesítményű, ún. just-in-time (JIT) fordítóprogrammal. Júniusban már elérhető volt a „beépített" Java a Microsoft Internet Explorer 3.0 web-böngészőjének bétaváltozatában.

LNX SecurNET — a „biztonsági háló"

A KFKI Számítástechnikai Csoport két hálózatos cégének (Lias, Networx) fúziójával létrejött Lias-Networx Hálózatintegrációs Kft. (LNX) a hálózatépítés minden „rétegében" dolgozik, az elődök valamennyi tevékenységi körét megtartva. A hálózati adatbiztonság igényeit felismerve, az LNX megalkotta saját SecurNET hálózatépítési koncepcióit, amely tapasztalatokból leszűrt tevezési és megvalósítási módszertan, kipróbált és bővülő elemkészlet együttese, mind arra a vezérelvre felfűzve, hogy a megfogalmazott hálózatbiztonsági igények a leghatékonyabban és a leggazdaságosabban legyenek kielégítve.

A SecurNET műszaki megoldásait tekintve elsőként az AT&T Systimax sodrott érpáras strukturált kábelezés jön számításba (amerikai FCC és európai előírásoknak megfelelően), illetve az üvegszálas kábelezés. Ezek elektromágneses sugárzást nem bocsátanak ki, tehát ellenőrzött körülmények között fokozott biztonságot nyújtanak. A SecurNET további termékeleme a WaveLAN szórt spektrumú rádiós kommunikáció, amely gyakorlatilag detektálhatatlan, ezért jól illeszkedik az LNX biztonsági filozófiájába. A LAN eszközök közül menedzselhető, biztonsági szolgáltatásokkal rendelkező termékek alkalmazásának támogatásáról van szó. A kapcsolási (switch) technológiák, sőt a virtuális LAN-ok alkalmazásával a biztonság szintje tovább fokozható (Cisco, Cabletron, HP, 3Com, Lucent Technologies eszközökkel). Az Internetbe szervezett LAN-ok védelmét különféle tűzfal-eszközök alkalmazásával éri el az LNX (ebben leginkább a Ciscóra támaszkodik). A hálózatmenedzsment a biztonság fontos összetevője. Az LNX e téren széles eszközválasztékból kínálhat, de leginkább a de facto ipari szabvány HP OpenView-t ajánlja, amelynek hivatalos VAR-ja ez a kft. (A Cabletron Spectrum is a kínálati palettán található.) Az LNX SecurNET koncepció megvalósításának eszközei a kriptográfiai hardver/szoftver megoldások (pl. CryptoCard, CryptoDisc stb.) A NetWare, Windows NT és Unix operációs rendszerek biztonsági funkciói szintén a módszertan lényegét képezik. Az LNX bejelentette azt is, hogy a kft. legújabb stratégiai partnere a Microsoft, amellyel Solution Partner-szerződést kötöttek a Windows operációs rendszerre és az Exchange levelezőrendszerrre vonatkozóan. Ez utóbbit a csoporttermékét támogató szoftverek között, az igen népszerű Lotus Notes mellett, annak alternatívájaként kínálja az LNX.

Kovács Attila

A szigetmegoldás nem (jó) megoldás

Sodorvonalban az Edifacttal

Sokan kérdezték, hogyan kapcsolódhatnának az ETR rendszerhez, vagy hogyan alkalmazhatnák ezt a technikát saját területükön? Nem olyan egyszerű erre a kérdésre válaszolni, de a pénzben is egyértelműen kifejezhető haszon miatt mégis megéri a fáradozást. Az ENSZ, az EU, a Magyar Tárcaközi Bizottság által ajánlott szabványokat még egy ideig ki lehet kerülni, de (hajós példánál maradva) aki szeretne a sodorvonalban maradni vagy oda kerülni, annak nem célszerű szigetmegoldásokban gondolkodnia. Szigetek pedig vannak szép számban.

Vannak olyan kisebb-nagyobb vállalatok, kórházak stb., ahol többféle rendszert használnak egyidejűleg, és arra törekednek, hogy egységes rendszert vezessenek be. Véleményem szerint nem biztos, hogy ez az egyetlen helyes út. Van helye a gigantikus alkalmazásoknak is, de a meglévő, bevált rendszerek integrálása is jó megoldásnak tűnik.

Ha például egy kórházat vizsgálunk, létezik járóbeteg-, fekvőbeteg- stb. rendszer. A röntgenosztályon a filmfogyást is figyelni kell, az ultrahangnál a képeket kellene tárolni, a laborban pedig soros porton kellene illeszteni egy mérőberendezést. A másik kórházban azonban rendszerint már nem ugyanilyet alkalmaznak. Azután, ott van a belső patikájuk, a bér és munkaügy, és még sorolhatnám. Persze, hallottam már integrált rendszerekről, de én nem nagyon hiszek bennük.

Szabad a gazda

Nem hiszek az integrált rendszerek rugalmasságában, jó árfekvésében stb. Sokkal inkább hiszek a kisebb, de önállóan is működőképes rendszerekben, amelyek viszont (az Edifact alkalmazásával) egymással szót értenek. Mindenki olyan programot vásárolhat, amely neki, a feladatnak és a pénztárcájának megfelel, vagy azzal készítteti el, akivel akarja. A vásárlónak és a felügyelő, irányító, tanácsokat adó szakhatóságoknak, szervezeteknek viszont arra kellene kínosan ügyelniük, hogy a rendszer képes legyen a számukra szükséges vagy előírt kommunikációhoz szükséges Edifact-üzenetek előállítására, fogadására, feldolgozására. (Az Edifact-üzeneteket szépen sorjában honosítani kell. Ezzel kapcsolatosan a Magyar Szabványügyi Testület ad felvilágosítást.)

Nézzünk egy egészségügyi példát. (Zárójelben az Edifact erre vonatkozó elektronikus szabványüzeneteit adom meg.)

Kezdjük a háziorvosnál, aki a beteget szakorvoshoz utalja. Ehhez (MEDREQ — orvosi szolgáltatás igénylése) üzenetet küld, mondjuk, a sebészetre. A kórház fogadó-elosztó számítógépe fogadja, és továbbítja az üzenetet a sebészet gépére, ahol az igényként jelenik meg.

Az adatcsomag tartalmazza a beteg és a vizsgálatkérés szükséges adatait. A beteg megjelenik, megvizsgálják, az adatokat kiegészítik, és továbbküldik a röntgenbe. A sebészet rendszere (Medreq — orvosi szolgáltatás igénylése) üzenetet indít a röntgennek. A röntgen gépe fogadja, a szükséges feladatok elvégzését követően (Medrpt — orvosi szolgáltatásról szóló leírás, Medruc — orvosi erőforrások használata, költség) üzenetet küld vissza a sebészetnek. A sebészet az üzenetet veszi, ha szükséges, kiegészíti és üzenetet küld (Medrpt — orvosi szolgáltatásról szóló leírás, Medrpe — egészségügyi előírás) a beküldő háziorvos számára.

A rendszerben keringő üzenetekből vagy annak kivonataiból kaphat másolatot az informatikai osztály, rajta keresztül pedig szintetizált adatokat kell kapjon a TB-OEP stb. Az OEP ma is kap adatokat, részben Dbase formátumban.

Azt és csak azt

A fenti példában bemutatott moduláris rendszernél nem lehet megmondani, hol van az informatikai rendszer határa. Az akár országos is lehet, nemzetközi és ágazatfüggetlen kapcsolatokkal. Az egyes kisebb-nagyobb rendszereknek lehetnek saját funkciói, adatállományai stb., amelyekre a másiknak nincs szüksége. Nem marad a felhasználóban olyan kétség, hogy esetleg más is beleláthat az adataiba, tudja, hogy az adatok tárolása sem terhel másokat, van, akinek a PC, van, akinek a Mac vagy a VAX, esetleg a „papné” tetszik stb. Az adatcsomagok (üzenetek) mozgása jól nyomon követhető, archiválható. Ha valamelyik rész leáll vagy éppen karbantartják, az nem blokkolhatja az egész rendszert.

De gondoljunk csak bele, ha kiállítunk valakinek egy számlát, vagy elküldünk egy megrendelést, mi sem írunk felesleges vagy nem oda illő adatokat a papírra. Ha valakinek a hitelkártyaszerű rendszer jutna eszébe mint alternatíva, nos, ez nem ez, hanem a rendszerben felhasználható technika mint azonosító, adathordozó, tárolóeszköz. Az MS Exchange vagy a Lotus Notes sem alternatíva, hanem szintén csak — akár többcélúan is felhasználható, rendszerbe integrálható — eszköz.

Partner és biztonság

Az ETR rendszer nem egy megalomániás számítógépes rendszer, amit rá szeretnénk erőszakolni mindenkire, hanem, ismételjük, megvalósult elemei mintaként szolgálhatnak.

Az adatcseréhez partnerre van szükség. Partner lehet a Bank, a Vám, a beszállító, a TB-nek a háziorvos, a patikák, a szakellátás, a kórházak, a háziorvosnak a szakellátás, az épülő logisztikai központok egymásnak és

partnereiknek, de akár egy nagyvállalat saját heterogén rendszerei is egymásnak stb. Számukra is rendelkezésre állnak már a szabványajánlások, csak alkalmazni kellene azokat, és nem újakat kitalálni, mint ahogy azt már sokan megpróbálták. Az Edifacthoz kapcsolódó szabványok egy részét már honosították, és MSZ szabványként megjelentek.

A kapcsolathoz kell egy partner (ha a rendszerben még nincs számára megfelelő), egy telefonvonal, modem és elektronikus postaláda (vagy más, elektronikus információtovábbításra alkalmas hálózat), kezdetben szaktanácsadó, esetleg oktatással támogatott tanulás, egy számítógép, és persze némi elszántság. Meg kell különböztetni azt, aki alkalmazni, és azt, aki fejleszteni szeretne.

Nem járatlan út

Az ETR (Egységes Telematikai Rendszer) már a negyedik sikeres projekten van túl, ahol (jórészt OMFB-pénzből) ajándékrendszereket kaptak az érintettek. Nemzetközi projektekkel is van kapcsolata. Ilyen példaképpen a Duna—Rajna—Majna-csatorna informatikai rendszere, vagy a COST 330-as projekt, amely az európai kikötők informatikai rendszerei közti kapcsolatteremtést célozza meg az Edifact alkalmazásával. Magyarországot is felkérték a részvételre, hiszen érdekli őket a keleti és a délkeleti irányú áruszállítás. Nyilvánvaló előny, ha ezek az informatikai rendszerek — bár heterogének —, mégis egységes rendszert képeznek az Edifact alkalmazásával. (A magyarországi COST 330-as projektről további részletek Tánczos Lászlóné tanszékvezető egyetemi tanártól tudhatók meg.)

Hogy ezek után ne tartson a kedves olvasó elvakult Edifact-hívőnek, megemlítem, hogy épp e cikk írásakor, azokban a napokban adtunk át egy olyan „adatátviteli" rendszert, amely két PC között úgy valósítja meg az egyirányú adatkapcsolatot, hogy az egyik vonalkódsorozatot nyomtat ki a másik számára, és a nyomtatványt az ügyfél viszi a második gépnek (Hegyeshalom határállomás, mérlegház —> spedíció). Szóval nem mondom azt, hogy az Edifact az egyedüli és üdvözítő megoldás, de a rendszereket tervezőknek, a fejlesztőknek, az informatikát oktatóknak és alkalmazóknak is ismerniük kellene ezt a szabványos technikát.

Találni képzetteket is

A BME nappali tagozatos képzésében mintegy 350 hallgató, a posztgraduális képzésben pedig mintegy félszáz hallgató részesül Edifact-oktatásban évente. Ha valaki bővebb információhoz szeretne jutni, akkor ajánlhatom neki az ENSZ-anyagokat (és a Hunpro által lefordított kiadványokat), az Edifact-tanfolyamokat, vagy pl. a www.altavista.digital.com kereső oldalán az Edifact szót. A www.editie.nl oldalakon szabad szoftverek és részletes dokumentációk is letölthetők. A kapott válaszözönön ne lepődjön meg senki, mivel ezt a technikát már sokan ismerik és alkalmazzák a világban.

A témában, ahogy az talán e cikkből is kitűnik, nem lehet szakmai féltékenységről beszélni, mivel az ily módon elkészült rendszerek jobbára erősítik egymás pozícióit.

Kiss János

Egységes, integrált felügyelet

A hálózatmenedzsment alapelvei

A hálózati menedzser feladata a hálózat folyamatos felügyelete és az esetleges beavatkozás, a hálózat 24 órás rendelkezésre állása a megfelelő szolgáltatási minőség érdekében. Amikor hálózatmenedzsmentről beszélünk, az esetek többségében az emberek általában egy drága szoftverrendszerre gondolnak, amely egy csapásra megoldja a hálózati menedzser (rendszergazda stb.) összes gondját, mivel (ahogy gondolják) csak rá kell kattintani a megfelelő ikonra a megfelelő képernyőn, és mindent azonnal meg fog oldani a szoftver...

A közhiedelemmel ellentétben hálózatmenedzsment az is, amikor a berendezések működőképességét például egy egyszerű IP ping paranccsal ellenőrizzük, és az esetleges hiba bekövetkeztekor az írott vagy a szájhagyomány útján terjedő „adatbázisból" kikeressük az adott rendszer fizikai és magasabb szintű, például szerverkonfigurációját, és ez alapján kezdünk bele a hiba behatárolásába, elhárításába.

Már ebből a példából is látszik, hogy a hálózatot használó összes berendezés, eszköz menedzsmentje fontos, mert csak a teljes rendszer egységes kezelése, egy felügyeleti rendszerbe való összefogása alapján tudunk gyors hibaelhárítást végrehajtani. A hálózat és a hozzá kapcsolódó berendezések egységes felügyeletét nevezzük integrált menedzsmentnek.

CMIS — az ISO ajánlása

Nem tudunk egységes menedzsmentstratégiát megvalósítani anélkül, hogy a hálózati berendezések menedzsmentje ne egységes alapelvekre épüljön. Hasonlóan, mint a hálózatok 7 rétegű OSI modelljére, az ISO kidolgozta az általános hálózatmenedzsment ajánlását is, CMIS (Common Management Information System — általános menedzsment-információs rendszer) néven. Mint azt a 7 rétegű OSI hálózati modellben tette, itt is alapelveket fogalmaz meg az ISO-ajánlás, amely alapelvek más, nem az ISO által kifejlesztett menedzsmentrendszerek számára is referenciának szolgálnak. A CMIS által definiált hálózati menedzsmentfunkciók a következő kategóriákba sorolhatóak (1. ábra):

Hibamenedzsment: segítségével észlelni lehet a hálózat bizonyos beállított paraméterektől eltérő viselkedését.

Konfigurációmenedzsment: segítségével az információátvitelt vezérelni lehet — az információ átvitelét végző berendezések paraméterezése, átprogramozása által.

Teljesítménymenedzsment: az adatfolyam forgalmi, késleltetési, torlódási, maradékkapacitás-adatait stb. gyűjti, ezek alapján a hálózat adatátviteli képességeit figyelemmel lehet kísérni. Ennek a funkciónak a része az adatok alapján történő tervezés- és kapacitásmenedzsment is.

Biztonságmenedzsment: funkcióegyüttes, amely szabályozza a hálózathoz, illetve a hálózati szolgáltatásokhoz való hozzáférés jogosultságait. Ez a funkció tartalmazza a hálózatmenedzsment-szolgáltatásokhoz való hozzáférési jogosultságokat.

Számlázásmenedzsment: a hálózati erőforrások felhasználásának adatait a megfelelő felhasználókhoz rendeli, miáltal lehetőség van a hálózati költségek megfelelő elosztására a felhasználók között.

Természetesen nem minden hálózati menedzsernek van szüksége valamennyi fent említett funkcióra. Például elképzelhető, hogy a számlázási funkcióra nincs szüksége, mivel nem végez belső elszámolást a vállalat belső szervezeti egységei között. Elképzelhető, hogy nincs szüksége a biztonsági menedzsment funkcióra sem, mert nincs külső belépési pontja a hálózatának, illetve a belső hozzáférési jogokat más szinten, például a szerverek hozzáférési jogosultsági rendszerén keresztül oldja meg. Azonban ahogy a hálózati rendszerek egyre bonyolultabbá válnak, ezek a funkciók is elérik, vagy akár meg is haladják a többi menedzsmentfunkció fontosságát és komplexitását, ezért gondolni kell megvalósításukra.

Nézzük meg tehát részletesebben, hogyan valósulnak meg egy rendszerben a sorolt funkciók.

Hibamenedzsment

Az egyik legrégebbi és legfontosabb funkció, amely általában minden hálózati menedzserrendszerben megvalósul. Alapeleme a hibamonitorozás, amely folyamatosan figyelemmel kíséri a menedzselt berendezés működését, és figyelmeztet, ha a berendezés működése eltér az előre beállított értékektől. Kétféle hibát különböztet meg. Kétállapotú hiba (hard error) az, amikor valamilyen erőforrás működése megszűnik, és ebből adódóan az adatátviteli kapcsolat megszakad, például egy 2 Mbit/s-os kapcsolat kiesik. Küszöbhiba (soft error) az, amikor valamilyen beállított küszöbértéket halad meg egy paraméter értéke, amit a menedzserberendezés jelez. Ez történik akkor, amikor például egy 2 Mbit/s-os összeköttetésben a bithibák aránya meghalad egy bizonyos, előre meghatározott szintet, ezért beavatkozást igényel. A hibajavítás egyrészt a menedzselt berendezések saját intelligenciája által valósulhat meg — például back-up route-ok (kisegítő adatátviteli utak) használata, spanning-tree bridge-ek (többszörözött hidak, amelyek közül egyidejűleg csak egy működik) alkalmazása —, vagy a berendezések konfigurációjának megváltoztatásával háríthatjuk el a hibát.

Konfigurációmenedzsment

A konfigurációmenedzsment vezérli a hálózati berendezések működését a beállított paraméterek megváltoztatása

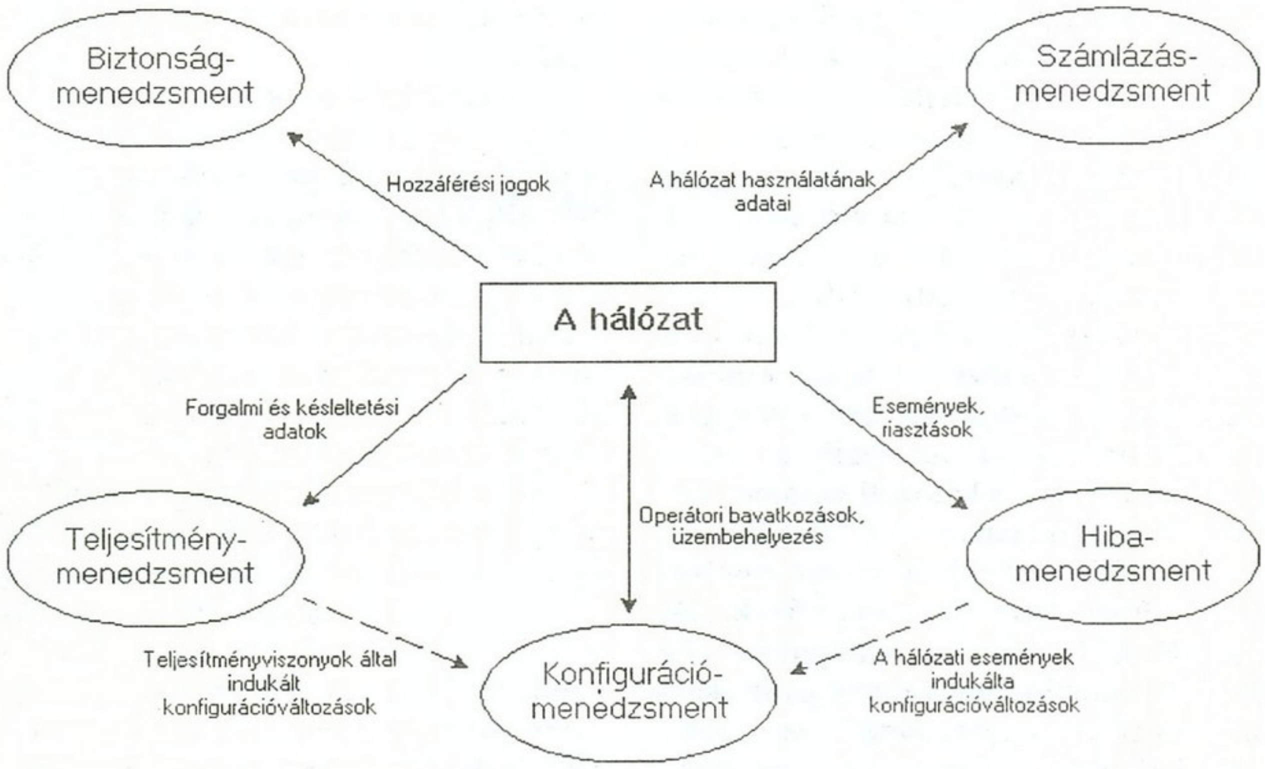
által. Ez gondoskodik a berendezések kezdeti és működés közbeni konfigurálásáról, ennek adminisztrálásáról, a hibás berendezések cseréje utáni konfiguráció letöltésről stb. Mint az előző pontban láttuk, a hálózat működése közben az adatátviteli paraméterek változása (hibaarány-növekedés, hálózati terhelés növekedése) a konfiguráció változtatását teheti szükségessé. Ez a konfigurációmenedzsment előre definiált eljárásai alapján, nagyobb rendszerekben mesterségesintelligencia-módszerek igénybevételével lehetséges, és több vizsgált bemeneti változó értékei szerint teheti meg a beavatkozást a hálózati berendezések konfigurációjában. (Például a Cabletron Spectrum hálózati menedzsment rendszere.) Ez különösen olyan komoly hálózatoknál fontos, amelyekben a nagy mennyiségű adatforrás, valamint a váratlanul bekövetkező eseményekre reagálás csak ilyen módszerekkel biztosítható.

Teljesítménymenedzsment

A funkció a hibamenedzsmenthez hasonlóan adott berendezések változóinak monitorozásán alapszik. Sok logikai hálózat egymás mellett létezése a hálózati forgalom ütközéséhez vezethet, aminek oka rejtve marad a logikai hálózatokat használók előtt, azonban ez a funkció segíthet felderíteni az esetleges lassulás okát. A teljesítménymenedzsment funkcióadatainak segítségével modellezhetjük a hálózat működését. Megbecsülhetjük egyes hálózati alkalmazások válaszidejeit, segítve ezzel a hálózati topológia, valamint a hálózati összeköttetések sebességének megtervezését.

A hálózat számlázásmenedzsmentje

A számlázásmenedzsment szükségessége, mint már említettük, elsősorban nagy hálózatok, több szervezeti egység közös hálózati erőforrásainak használatakor jelentkezik. Többféle paraméter mérésére is szükség lehet a hálózati erőforrások igénybevételének függvényében. Például csomagkapcsolt hálózatok esetében a kapcsolatfelépítések száma, az átvitt csomagok mennyisége, a virtuális kapcsolatok fennmaradásának ideje mind költségtényező lehet, tehát mindegyik méréséről, a forgalmi adatok felhasználókhoz való hozzárendeléséről gondoskodni kell. Szükség lehet rá, hogy a hálózat által biztosított szolgáltatások (mail, directory services) igénybevételét szintén valamilyen módon a felhasználáshoz



rendeljük, esetlegesen ennek méréséről is gondoskodni kell.

Biztonságmenedzsment

A biztonság szó ebben az esetben a hálózati erőforrások illetéktelen manipulálástól való megvédését jelenti. Ez nagy hálózatoknál nagyon bonyolult és szerteágazó feladat lehet. A felhasználók elsősorban a számítógépeken tárolt adatok megszerzésében, manipulálásában érdekeltek, ezért ezeket magasabb szintű hozzáférési eljárásokkal védik. Ide tartozik a hálózat távoli felhasználóinak megadott belépési jogosultságok menedzselésétől kezdve az Internethez kapcsolódó hálózatok tűzfalakkal történő megvédésének felügyeletén keresztül egészen a hálózatimenedzser-állomások hozzáférési jogosultságainak ellenőrzéséig minden tevékenység.

További szabványok

Az előző funkcióleírásból is sejthető, hogy egy jól kezelhető menedzserszoftvernek milyen bonyolult követelményeknek kell megfelelnie. Természetesen manapság a grafikus felhasználói interfész, valamint a modern relációs adatbáziskezelés is tartozéka a hálózati menedzsment rendszereknek. A nagy, több telephellyel rendelkező vállalatok menedzsment-igényeinek kielégítése céljából ún. elosztott (distributed) mendzsment-rendszereket is létrehoztak már. Ezek a szoftverrendszerek lehetővé teszik, hogy a hálózat elosztott eroforrásait elosztottan felügyeljük, azonban egyes hálózati funkciókat képesek legyünk központilag is elérni (például számlázás, konfigurációmenedzsment stb.).

Az ISO CMIS modell ismertetése után nézzük meg, milyen hálózati szabványok terjedtek el a gyakorlatban.

CMIS/CMIP

A CMIS rendszer kezelésére az ISO az ún. CMIP-t (Common Management Information Protocol — általános menedzsment-információs protokoll) definiálta. A CMIS/CMIP a mendzserközpontokból, valamint a menedzselt objektumokból, vagyis azokból a berendezésekből áll, amelyek megértik a CMIP protokollt. A menedzselt berendezéseken a paraméterek ún. MIB-eken (Management Information Base — menedzsment-adatbázis) keresztül érhetők el. A CMIS/CMIP definíció objektumorientált, ami különösen nagy rendszerek felügyeletében jelent előnyt. A CMIP protokoll implementálása igen bonyolult feladat, a menedzselt berendezésektől nagy számítási kapacitást, sok memóriát igényel. Ezért nem meglepő, hogy elsősorban a legnagyobb hálózati rendszereket gyártó vállalatok hozták ki a CMIS/CMIP protokollon alapuló rendszereket. (Pl. AT&T, British Telecom stb.) A rendszer a számítási teljesítmény és a memóriaárak csökkenésével várhatóan mind elterjedtebbé válik.

SNMP és SNMP version 2

Az SNMP (Simple Network Management Protocol — egyszerű hálózati menedzsment protokoll) népszerűségét a TCP/IP hálózatok széles körű elterjedésének, valamint igen egyszerűen implementálható protokolljának köszönheti. Eredetileg a TCP/IP hálózatokban használatos gateway-k mene-

dzelésére fejlesztették ki Simple Gateway Management Protokoll néven, csak később terjesztették ki használatát más hálózati berendezésekre is.

A protokoll kétféle résztvevőt definiál. Egyrészt menedzserállomást, másrészt menedzselt (ún. ügynök-) állomást, amely a menedzserállomás kérdéseire válaszol. Az ügynökállomás paraméterei az előző pontban említett MIB-ekben tárolódnak, amelyek speciális fastruktúrába szervezik a menedzselt berendezés hálózati paramétereit tartalmazó adatbázist. Az adatbázisnak vannak előre definiált területei, amelyeket minden berendezésgyártónak, aki az SNMP protokollt támogatja, meg kell valósítania, míg vannak gyártóspecifikus részei is, amelyek csak a gyártóra jellemző adatbázist tartalmaznak.

A protokollról

A protokoll előnyei, nevezetesen az egyszerű megvalósítás és architektúra egyben a legfőbb hátrányai is. Az egyszerű protokoll könnyű lehallgathatóságot is jelent, a kérdés-feleleten alapuló architektúra nagyobb hálózatokban egyre növekvő hálózati forgalmat generál.

Többek között ezeknek a problémáknak a megoldására fejlesztették ki az SNMP version 2 protokollt, amely az SNMP előnyei megtartásával próbálja annak hátrányait kiküszöbölni, tehát megoldja a lehallgatásból adódó problémákat, biztosítja két menedzserállomás közti adatátvitelt, segítséget nyújtva nagyobb elosztott menedzserrendszerek kialakításához stb.

Az SNMP protokoll nagy sávszélességigényének csökkentésére dolgozták ki az ún. RMON (Remote Monitoring — távoli monitorozó) funkciót. Az RMON funkció például egy router berendezésben az SNMP protokoll szempontjából egy agent funkciót megvalósító eszközben került definiálásra. Feladata az, hogy egy adott interfész például a helyi LAN forgalmát figyelemmel kísérje, a forgalmi jellemzőket az időinformációkkal együtt feljegyezze, tehermentesítve ezzel a menedzserállomást, valamint a távoli router és a menedzserállomás közti esetlegesen kis sebességű soros összeköttetést az egyébként szükséges SNMP-forgalomtól. Az adatokat az RMON számára definiált MIB struktúrában tárolja. A menedzserállomás az őt érdeklő adatokat az SNMP protokoll segítségével képes ebből az adatbázisból kinyerni, és ennek alapján megjeleníteni, eltárolni stb.

A fejlődés iránya

A ma használatos legtöbb hálózati menedzserszoftver, valamint hálózati berendezés elsősorban az SNMP protokollt támogatja. Várhatóan az elkövetkező néhány évben még nem változik a helyzet, mivel az SNMP version 2 felhasználása most kezdődik. A version 2 ugyanis természetes továbblépés, szemben a más logikájú CMIS/CMIP architektúrával. Tehát egyelőre egyeduralkodók lesznek az SNMP-s rendszerek, egészen a csak néhány hálózati funkciót megvalósító PC-s menedzsmentszoftverektől a sok gyártót támogató, RISC-es számítógépeken futó, nagy menedzserplatformokig (Novell NMS, HP Open View, SunNet Manager, Cabletron Spectrum stb.).

Menyhért Zoltán

Ha van egy jó felügyelő...

„Valaki" odafigyel a hálózatra

Nemrég még a számítógépre mondhattuk, hogy egyre több területen, egyre több helyen nélkülözhetetlen. A jelzőt most már a hálózatra is kezdhetjük alkalmazni, mert közeledik az idő, amikor csak az érheti el gazdasági, tudománybeli céljait, aki hálózatba kapcsolt számítógépet használ.

A következő években robbanásszerű terjedése várható azoknak a rendszereknek, amelyek egyszerre több embert szolgálnak ki, amelyekben osztott vagy centralizált adatbázisban, de együtt van mindaz az információ, amely ma még elszigetelten rejtőzik a munkatársak számítógépében vagy netán a fejében.

A hálózatosodás azonban kiszolgáltatottságot is jelent. Ha minden adat bevonul a gépbe, akkor elég egy áramszünet ahhoz, hogy megbénuljon az élet az irodában, s ez komoly veszteséget okozhat — arról nem is szólva, hogy mennyire megviseli az embereket, ha képtelenek valamit határidőre elvégezni. A hálózatosodásnak tehát alapvető feltétele a megbízhatóság, a lehetőség szerinti hibamentes működés, illetve a mégis bekövetkező hibák gyors elhárítása.

Ehhez azonban meg kell találni a hiba okát, ami bizony annál nehezebb, minél bonyolultabb egy rendszer — márpedig egy hálózat már meglehetősen bonyolult. Ráadásul négy-öt-tíz PC-ből álló kis hálózat nem nagyon tud eltartani egy olyan szakembert, akinek nincs más feladata, mint a hibaelhárítás, a hálózat gondozása. Szükség van tehát — és egyre nagyobb szükség van — a hálózat működésének automatizált megfigyelésére, esetleg némi önjavító képességre is.

A centrumban

Az általánosan használt, szabványosodó hálózatmegfigyelési modellek alapeleme a megfigyelt objektum, a saját működéséről a hálózaton keresztül jelentést adó berendezés vagy alkalmazás, illetve ezek egy-egy csoportja. A modellben adjuk meg az objektumok tulajdonságait és a hozzájuk tartozó műveleteket. A hálózatfelügyelet kialakításához hozzátartozik egy állandóan futó felügyelőprogram, processz létrehozása, amely futhat a szerveren, de futhat egy külön számítógépen is. Ez gyűjti össze a végpontokon, aktív hálózati elemeken dolgozó ügynökök által érzékelt eseményekről, állapotváltozásokról beküldött adatokat.

Tekintve, hogy a hálózatfelügyelő program általában magas szintű alkalmazás, jól meghatározott, stabil protokollokkal dolgozik, például TCP/IP-vel. Számos modell készült, közülük a legátfogóbbak az Internet protokollra és az ISO hétszintes OSI — open systems interconnection — szabványára épülnek. Vezető megoldásnak számítanak azok is, amelyek nagyszámítógépes környezetben alakultak ki, mint az IBM, a DEC vagy az AT&T által ajánlottak.

SMI + MIB + SNMP

Az IP modell egyik fő része az SMI — structure of management information: a felügyelethez szükséges, strukturált információ —, amely hierarchikus szerkezet, levelein a hálózat alapvető jellemzőivel, a köztes szinteken pedig azokkal az objektumokkal, amelyek állapotának a leírásához már több jellemzőre — alárendelt objektumok állapotának az áttekintésére — van szükség.

Az IP modell második fő fogalma a MIB, vagyis management information base, amely nem más, mint ennek az információs fának egy ága, ebbe kerülnek működés közben a hálózat részelemeinek megfigyelt paraméterei, s innen kérheti le azokat a felügyelőrendszer egy ügynök segítségével.

A harmadik, és legszélesebb körben ismert része az IP modellnek az SNMP, vagy simple network management protocol. Az SNMP-ügynök figyeli meg a hálózati berendezést, s továbbítja — folyamatosan vagy megkeresésre — a kapott adatokat egy felügyelőállomásra. Erről a felügyelőállomásról juttatják el a már említett felügyelő ügynökök (agents) az információt a felügyelő processzhez, amikor szükség van rájuk.

A legújabb szabványokban az SNMP állomáshoz és az ott gyűlő adatokhoz sokfelől hozzá lehet férni, vagyis az SMI fában az nem csak szigorúan egyféle környezetben, a MIB alá rendelve szerepelhet. Fontos szerep jut az IP modellben az SNMP-ügynök által a felügyelőállomásnak küldött figyelemfelhívó adatcsomagnak (trap), amelyet az minden olyan esetben elindít, amikor valami szokatlant észlel.

A jóval összetettebb OSI modellt elsősorban az különbözteti meg az IP-től, hogy egyre többet vitatkoznak rajta, míg az IP-t egyre több helyen alkalmazzák. Mindazonáltal jóval bővebbek a lehetőségei, mint az IP-nek, például a

MIB-jébe több információ fér, és rugalmasabb is, mint az IP modellbeli. Öt fő tartományát, rendszerfelügyeleti funkcióterületét határozzák meg benne a felügyeletnek: a könyvelést, a konfigurációt, a hibakezelést, a teljesítmény és a biztonság felügyeletét. Érdemes még megemlíteni az IBM NetView-n keresztül megvalósítható network management architecture-t, amely a systems network architecture kiterjesztése, valamint a digital enterprise management architecture ajánlását.

ManageWise 2.0

Az IP modell egyik legteljesebb inkarnációja a Novell és az Intel közös rendszere, a ManageWise 2.0. A ManageWise-t minden lényeges képességgel felruházták, ami a mindennapos hálózatfelügyelet ellátásához szükséges. Ötvözi a Novell felügyelőrendszerének, az NMS-nek, a NetWare felügyelő és LANalyzer ügynökének technológiáját az Intel LANDeskjének felügyelő és vírusvédő szolgáltatásával. A teljes hálózat megfigyelésére képes — a felhasználótól a szerverekig és a hálózati infrastruktúráig —, és tartalmaz eszközöket a beavatkozásra is.

Az IP modellek lényeges részét képező SNMP mellett támogatja a távoli monitorozás RMON szabványát is. Speciális részfeladatok megoldására számos külső fejlesztő készített és készít alkalmazást, amelyek beillesztését, használatát a ManageWise jól támogatja.

Ha tennie kell valamit...

Melyek azok a hálózati problémák, amelyek megoldásában számíthatunk a felügyelőprogramra? Sokszor vannak gondok a hálózati nyomtatással. Kifogy a papír, beszorul a papír, egy elindított nyomtatás túl sokáig várakozik a sorban, és megelőzik az utána jövők. Ha a hálózati nyomtató tud SNMP-jelentést adni, és van hálózatfelügyelő program, akkor jóval hamarabb kiderül a baj, gyorsabban lehet intézkedni.

Előfordulhat, hogy a felhasználó egyszer csak nem fér hozzá egy dokumentumhoz, egy hálózati alkalmazáshoz, amellyel pedig korábban már dolgozott. A ManageWise-ba integrált NetWare adminisztrációs eszközökkel ennek oka gyorsan felderíthető, s ha nem azért történik minden, mert túl sokan vannak bejelentkezve a szerverre, hanem azért, mert valami változás van a rendszerben, akkor pillanatok alatt intézkedni lehet. A felügyelőprogram által összegyűjtött adatok elemzésével gyorsan felfedezhető, ha valahol gondok vannak a válaszidővel, s a lelassulás oka is kideríthető.

Sokkal összetettebb problémák jelentkezhetnek egy hálózati szervernél, előfordulhatnak hibajelenségek, amelyeknek a gyökerét egészen máshol kell keresni. A mintegy 400 paramétert folyamatosan lekérdező, és a beépíthető korlátokkal ütköztető felügyelőprogram jó előre jelezni képes, ha a szerveren kevés a szabad hely, ha egy állományt nagyon sokszor kérnek, s ezért érdemes áthelyezni, meggyorsítani a kiolvasását. Figyelmeztet a forgalom növekedése nyomán indokolttá váló CPU-cserére vagy tárbővítésre.

Az automatikus térképezés, leltározás segítheti a hálózat áttekintését, és ennek révén azonnal kiugrik, ha valamelyik gépbe belenyúltak, kicserélték benne a processzort vagy a merevlemezt. A részletes leltár a karbantartást is megkönnyíti. Nemcsak a hibafelderítés a hálózatfelügyelet dolga, hanem annak ellenőrzése is, hogy egységes-e a hálózat, a kezelők nem összevissza választottak-e szoftvert, és mennyire egységesek azok a programok, amelyekkel sokan dolgoznak. Erre a legjobb megoldás a központi szoftverszétosztás, amit például a ManageWise is támogat.

Reagálás a jelzésre

A ManageWise-zal felszerelt hálózatban a felügyelet lehet teljesen automatizált, amikor a hálózatintézőnek csak néha, illetve hibajelzéskor kell ránéznie a konzolra. Éppen ezért lehetőség van arra, hogy a felügyelet máshol történjék, valahol távol a hálózattól. Különösen a kisebb — 5–25 gépből álló — hálózatok működését érdemes így figyelni. A cég, amely egyszerre sok helyi hálózat felügyeletét el tudja látni, kapcsolatba léphet azokkal rendszeres időközönként igénybe vett kapcsolt vonalon, de természetesen lehetőség van az állandó összeköttetésre is. Riasztáskor a helyi ManageWise azonnal küldi a jelzést a felügyelőprogramnak, s ott az ügyeletes szakember intézkedhet.

Van ugyan néhány jelzés, amelyet nem is érdemes továbbítani — például hogy kifogyott a nyomtatóból a papír —, vagyis a jól kialakított távoli felügyeletnél is vannak helyben elvégzendő, komolyabb szakértelmet nem igénylő feladatok. Azért is érdemes megszűrni a távoli felügyeletre bízott feladatokat, mert a vonali kapcsolat pénzbe kerül.

De egy lépéssel tovább is mehetünk: a mobil kapcsolat révén a távoli felügyeletet akár egy rádiótelefonos noteszgép is elláthatja, vállalkozó kedvű szakemberek ennek birtokában elvállalhatják akár tíz-húsz kisebb helyi hálózat felügyeletét is, és e rendszer működtetési szisztémája egyre inkább közelíteni fog a telefonéhoz.

Vargha Márton



AZ INTERAKTÍV MÉDIÁK MAGAZINJA

Multimédia Magazin Mindenkinek

Az idei második szám megjelenik június végén

A tartalomból:
Mérlegen a hazai lexikon-CD-k, Monitorok óriástesztje, Slide show a karosszékből Portfolio Photo-CD-vel, Trükkök a CorelDRAW-val, Video a multimédiában, Audio-CD-válogatás, Ismerkedés a CompuServe-vel, Hálózati multimédia-fejlesztők és bemutatkozik többtucatnyi CD-újdonság.

A CD-MELLÉKLETEN

Mallorcai utazás, Soproni körkép, Lánchídi hangulatok, Dolák-Saly Róbert videoklipje, Az év fotói 1995, Multimédia-iskola, A tokaji bor apoteózisa, Új hazai CD-k demói és shareware csokor.

**A CD Panoráma megjelenik évente négyszer, előfizethető a Computer Panoráma Kiadónál.
cím: Computer Panoráma Kiadói Kft.
1091 Budapest, Üllői út 25. II. emelet
Telefon: 218-3011/302, fax: 217-2646
Teljes éves előfizetés esetén ajándék CD-box!**

MEGRENDELŐLAP
Előfizetéssel megrendelem 1996-ra a CD Panorámát
(négy szám ára: 3920 Ft, az idén még hátralévő két szám ára: 1960 Ft)

Név: ....

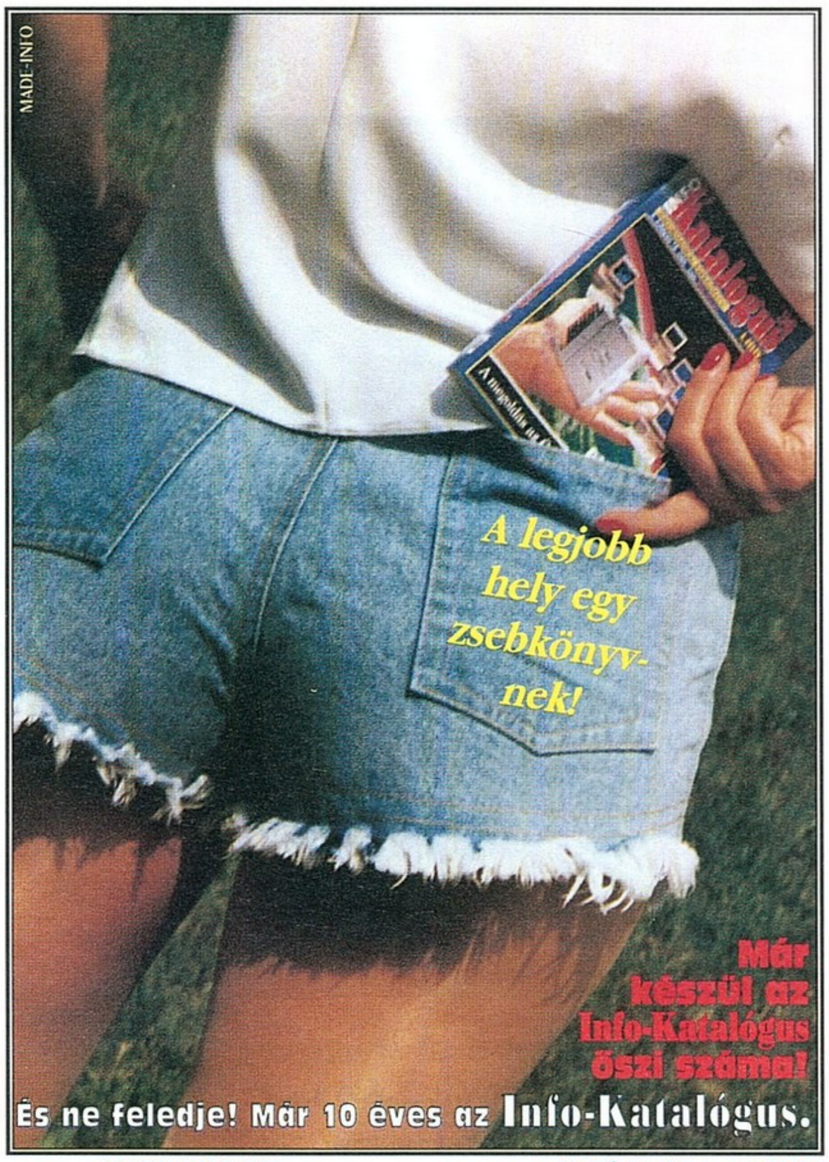
Postacím: ....

Telefon: ....

Olvasható aláírás: ....

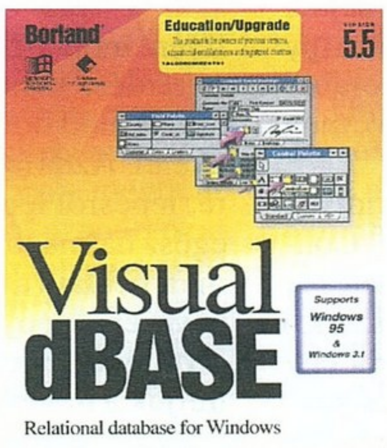
LAPJAINKKAL CÉLBA TALÁL

INFORMÁCIÓKÉRÉS: 0704 ▼



Kiadó : MADE-INFO KFT. *Levélcím* : 1506 Budapest Pf. 99
Telefon : 227-3647, 246-2735 *Telefax* : 228-1934, 246-2735

INFORMÁCIÓKÉRÉS: 0716 ▼



Intranet kiegészítések a Visual dBASE-hez

Visual dBASE Web expertek
DeltaPoint WebTools
Web Server támogatás
HTML engine
CGI osztálykönyvtár
Egyszerű programozás
CGI mintaalkalmazások

mindez 4.900,- Ft + ÁFA

És ha a szoftver még hiányzik:
Visual dBASE kompetitív upgrade 9.900,- Ft + ÁFA
Delphi 1.0 Desktop kompetitív upgrade 14.900,- Ft + ÁFA

SilVert

Borland Delphi-Szoft **Delphi-Szoft**
1085 Budapest, Horánszky utca 26.
Telefon: 138-4144, fax: 118-0915
Internet: http://www.delphi.hu

INFORMÁCIÓKÉRÉS: 0706 ▼

Sejtek, „automatizált" sejtek, sejtautomaták

Mesterséges élet?

Mint tudjuk, szervezetünk sok apró sejtből áll. E sejtek együtt nagyon összetett műveletek elvégzésére képesek. Egy-egy sejtre e műveleteknek azonban igen kis része eshet. Innen jött az ötlet, hogy sok-sok, ám önmagában egyszerű műveletet elvégző részből próbáljunk meg komplex rendszereket előállítani. Van még egy fontos megkötés, hogy ezek a részek azonos tudásúak legyenek, azaz egyik se specializálódjon valamire, ellentétben a neuronhálók elemeivel.

Az életfolyamatokra hasonlító működés utánzásakor leggyakrabban egy diszkrét modellt használunk. Ez azt jelenti, hogy egy (esetleg speciális) háló rácspontjaiban állnak azok a részek, amelyeket az egyszerűség kedvéért sejteknek nevezünk. Ezek a sejtek végesen sok állapotot vehetnek fel. A sejtek állapotai lépésről lépésre változnak, így a sejt és a környező sejtek korábbi állapotainak függvényében kapjuk meg a sejt új állapotát.

Hogy ez érthetőbb legyen, tekintsük a legismertebb sejtautomatát, a Conway-féle életjátékot. (Erről részletesen szól a rovat másik cikke. — A szerk.) Itt minden sejt élő vagy halott. Ha egy élő sejtnek két vagy három élő szomszédja van (a nyolcból), akkor továbbra is életben marad, máskülönben kipusztul. A halott sejt életre kel, ha pontosan három élő szomszédja van. (Ezt az életjátékot választotta képernyővédőjének a Sun, és rátettük a lemezmellékletre is az egyik verziót, de erről még külön is szólunk.)

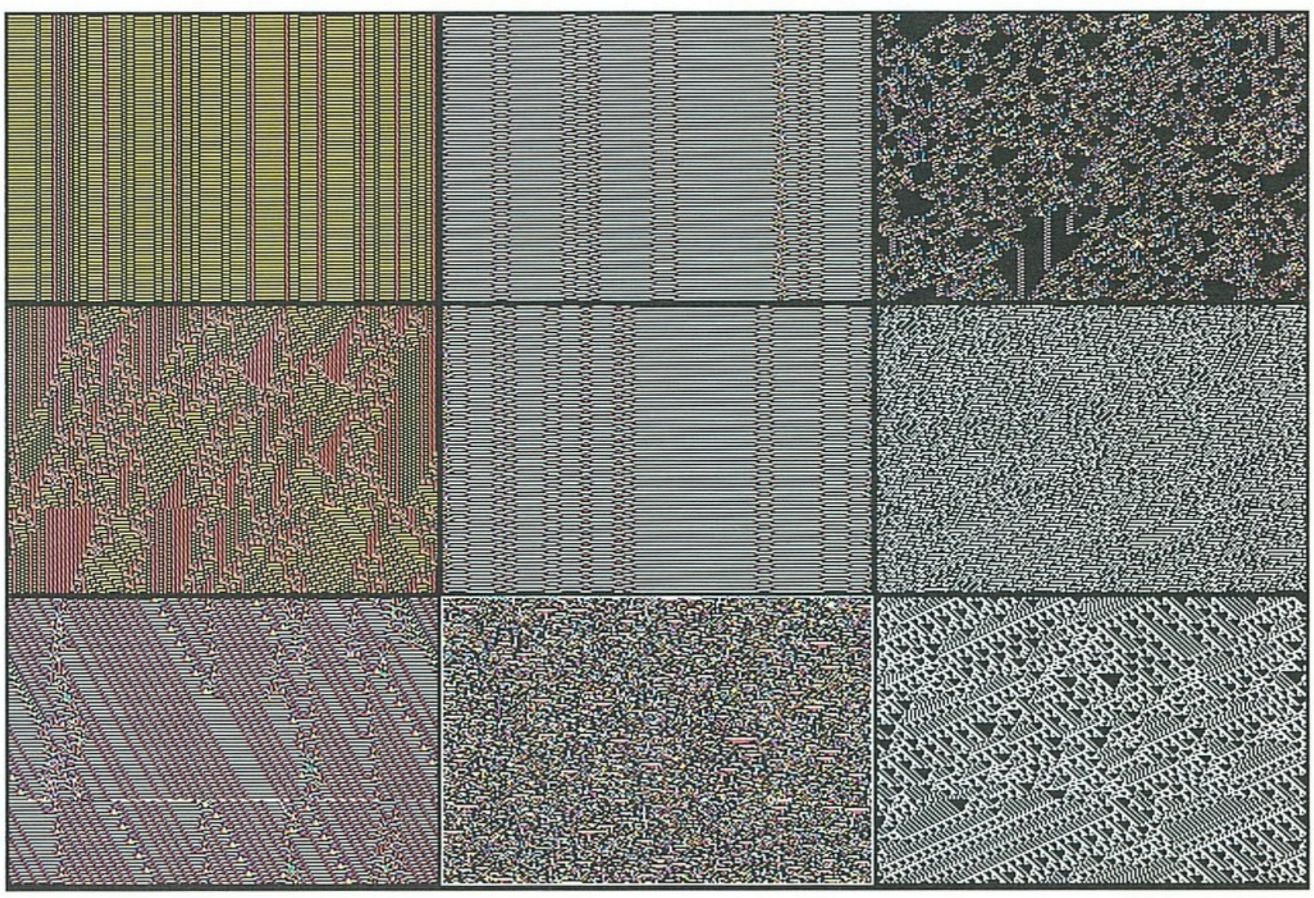
Ezzel a programmal találkozva még senki nem vár sokat a sejtautomatáktól. Ugyanakkor sejtautomatákkal megvalósíthatjuk például az AND, OR, NOT logikai áramköröket, sőt a memóriát is. Ez pedig azt jelenti, hogy amit ilyen elemekből össze lehet rakni (mondjuk egy processzort), azt sejtautomatával is megoldhatjuk.

Dinamikus rendszerek

Persze nem valószínű, hogy bárki is ezentúl mondjuk C++ helyett életjátékot fog használni programozási nyelvnek. Viszont vannak olyan területek, ahol érdemes sejtautomatákat használni. Ilyen lehet a dinamikus rendszerek vizsgálata, mint például a fűfajták versenye a területért; az erdőtűz terjedése; a plankton szaporodása a tengerben. Meg lehet próbálkozni az időjárás vagy akár a Nap szerkezetének leírásával. Ilyen modellek felállításakor felmerül a kérdés: mennyire teljesülnek a modellezendő dolognál a sejtautomata feltételei. Azaz mennyire történik egyszerre, lépésről lépésre (diszkrét idő), és az egész halmazon azonosak-e a feltételek, mert például a mélyebben lévő plankton kevesebb fényt kap, mint a felszínen lévő. Megtehetnénk, hogy tovább finomítjuk a modellt, de összhangban lesz-e a modell differenciálódása a számolás bonyolódásával?

A sejtautomaták esetén, ha tudni szeretnénk, hogy mi lesz az automata állapota (vagy akár egy sejt állapota) 25 lépés múlva, akkor mivel ide csak az ezt megelőző 24 lépésen keresztül juthatunk, ki kell számolni minden egyes sejt állapotát minden egyes lépés esetén. Ez egy méretesebb sejtautomata esetén tekintélyes számolást jelent. Ki lehet próbálni a lemezmellékleten szereplő programmal, hogy elegendő egyetlen sejt állapotát megváltoztatni, és ez maga előtt görgeti a változások sokaságát. Egy Kínában röpködő pillangó szárnycsapásai vihart okozhatnak Angliában. Gondolom, mindenki felkapta a fejét, amikor a káosszal kapcsolatban erről először hallott. (Alaplap, 1993. június.) De hát ez is épp ugyanaz, csak másik oldaláról közelítjük meg a dinamikus rendszereket. Mivel bizonyos szabályok esetén ennyire instabil lehet a sejtautomata, ember legyen a talpán, aki megmondja, honnan indultunk, mi volt 25 lépéssel korábban.

Látjuk, hogy van egy remek módszerünk titkosításra, elegendő a jelszókérésnél a kapott karaktersorozat valamilyen kódolt alakját odaadni a sejtautomatának, végezzen el 10 lépést, és hasonlítsa össze a kapott eredményt a már



tárolttal. Vannak olyan szabályok is, amelyek megfordíthatók, így ha ilyen szabályt alkalmaz a sejtautomata, akkor a kódolt üzenetet vissza is nyerhetjük.

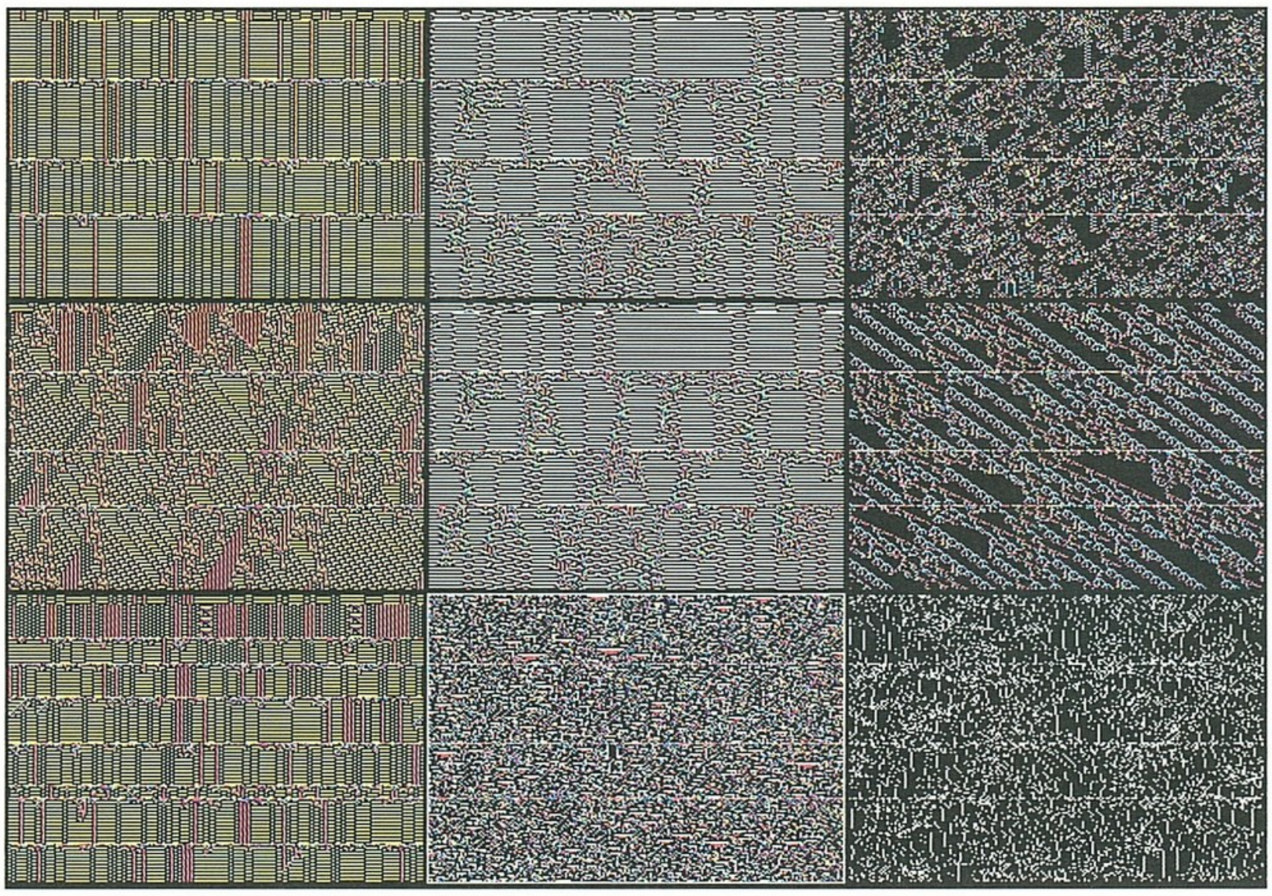
Dimenziók — praktikusan 3-ig

Többfajta sejtautomata létezik. A legegyszerűbb közülük (amely használható is valamire) az egydimenziós; itt az automata egy állapota egy sort foglal el, és az egymás után következő állapotokat egymás alatti sorokkal ábrázolhatjuk. Ezt az automatát a legegyszerűbb programozni, és itt a legkönnyebb szemmel követni a változásokat. Ennél már bonyolultabb a kétdimenziós automata, ahol egy-egy képernyőt foglal el egy-egy automataállapot. Ennek programozásához mindenképpen kell egy képernyőnyi segédtömb, amelyet az új állapot kiszámításához használunk. Ilyen a Conway-féle életjáték is. Lényegében korlátlanul növelhetjük a dimenziószámot, bár szerintem már a háromdimenziós sejtautomatákat is nehezen követhetjük szemmel, noha ezekkel még érdemes foglalkozni, hiszen a körülöttünk lévő világ is háromdimenziós.



A Calife programmal egydimenziós sejtautomatákat próbálgathatunk. Itt nem adatott meg számunkra, hogy az automatát kedvünk szerint programozhassuk. Beleszólhatunk, hogy hány különböző állapota lehessen egy sejtnek, vagy hogy a sejt állapotát hány szomszédja határozza meg. Emellett megadhatjuk azt is, hogy a sejtautomata szabályaiban mekkora arányban szerepelhet a 0 érték. Mivel életünkben igen fontos szerepet játszik a szimmetria, itt is megadhatjuk, hogy szabályaink csak szimmetrikusak legyenek, de ha így nem kapjuk meg a várt eredményt, egy kis mutációval segíthetünk összeborzolni a rajzokat. A program használatához legfontosabb megjegyezni az F1 billentyűt, mivel erre megkapjuk a használt billentyűk listáját. Ha valamely sejtautomata megtetszett, akkor annak a szabályát lemezre menthetjük, és bármikor visszatölthetjük a képernyőn szereplő kilenc sejtautomata bármelyikének helyére.

A WireWorld programot elindítva elektromos kapcsolásokkal szórakozhatunk. A dokumentációban megtalálhatjuk azokat az egyszerű szabályokat, amelyek alapján ez a sejtautomata működik. Bármikor belerajzolhatunk a képernyőn lévő ábrába, és a következő lépésben ezt már figyelembe is veszi az automata. Itt is segítséget kérhetünk az F1-gyel. A demóprogramokban megtalálhatjuk a legelemibb logikai kapcsolásokat. Ha képernyőnk bírja, akár 50×80-as ábrákkal is dolgozhatunk, s ha ez sem elég, akkor a mellékletként adott forrásprogramot át kell írni nagyobb képernyőméretre.



Amerikai magyar és társai

Utolsónak lássunk egy magyar programot is. (Jellemző, hogy ezt a programot először egy amerikai gépen találtam meg.) Ha több különböző sejtautomatát akarunk elkészíteni, akkor újabb programokat kellene írni, és ezeket sorra lefordítani. Egy kis változtatás miatt kellene elővenni a fordítót. A szerzők azt a szerencsésebb megoldást választották, hogy írtak egy keretprogramot, amelyben menet közben egyszerűen megadhatjuk a szabályokat. (Ebből is láthatjuk, hogy a programozható eszközök használata sokkal kényelmesebb a lebetonozott programoknál.) Ez a keretprogram egy Forth interpreter lett, mely azonban nálunk minden előnye dacára nem igazán terjedt el. A példaprogramokból és a szűkös leírásokból egy kis munkával azonban megismerhetjük ezt a nyelvet, és a számunkra fontos sejtautomatákat könnyedén megírhatjuk. Sajnos ez a program a GNOM projekt keretében íródott, így igen primitív a külalakja, de az a lényeg, hogy használható! Az első indítás előtt érdemes elolvasni az angol nyelvű leírást.

A sejtautomaták változatai remek szórakozást is nyújtanak. Egyszerű szabályokkal (így tehát könnyen progra-

mozható) látványos programokat készíthetünk. Lássuk például a hangyát. Ez elindul a rácson (amit véletlenszerűen befestettünk), és ha nem fehér mezőre lépett, azt pirosra festi, majd balra fordul; ha piros mezőre lépett, akkor azt fehérre festi, majd jobbra fordul. Ezek után az adott irányba lép egyet. Ez az algoritmus könnyen programozható. Ha akarjuk, írjuk át a szabályokat, vagy akár eresszünk egyszerre több hangyát is a rácsra.

Látványos lesz az a program is, ahol egy sejtnek nem négy vagy nyolc, hanem hat szomszédja van. Ezt sem nehéz ábrázolni, elegendő minden második sort (vagy oszlopot) egy fél egységgel eltolni a szomszédos sorokhoz (vagy oszlopokhoz képest). Ha egy sejtet két egymás mellett lévő pixellel jelölünk, ez egypixelnyi eltolást jelent. Ha szabályaink szimmetrikusak, és időről időre megváltoztatjuk őket, akkor gyönyörű hókristályokat rajzolhatunk a képernyőre.

Egyszerűen programozható az a feladat is, amikor egy füves mezőn nyulak és rókák élnek. Ha egy adott sejt közelében nincs sok nyúl, akkor ott a fű vidáman zöldell. Ha a környéken nincs sok róka, akkor nyugodtan szaporodhatnak a nyulak, míg ha valahol sok a nyúl, akkor ott egy róka is elélhet. Ha pedig valahol kevés az élelem, ott az állatok éhen halnak. Próbáljunk ezek alapján olyan szabályrendszert összeállítani, hogy a szabályos életciklusoktól eltekintve azonos legyen a populáció mérete, azaz egyik állatfaj se haljon ki. (Környezetvédőknek hatásos eszköz lehetne ez a feladat, hogy megmutassák: mennyire képes kordában tartani az ember az általa „megszelídített" környezetet.)

Egyesek az életjátékot úgy változtatták meg, hogy táblás játék lett belőle: a játékosok egymás után tesznek egy-egy saját sejtet a táblára, és minden lépés után újragenerálódik a sejtautomata. Az veszít, akinek eltűnik az összes sejtje.

Felkészítő...

Lássunk néhány tanácsot, amelyet érdemes megfontolni saját programok írásakor! Ha egymásba ágyazott sima ciklusokat használunk, akkor ugyanazt a sejtet igen sokszor kell újraolvasni, ezért inkább kódoljuk le úgy, hogy a következő sejt környezetének állapotát az előző sejt környezetének állapotából bitmanipulációkkal (eltolás stb.) lényegében megkapjuk. A szabályokat inkább táblázatban tároljuk, mintsem beprogramozzuk, s így már egyetlen művelettel megkaphatjuk a sejt új állapotát. Feltétlenül két rácsot kell használni, egyiken a régi állapot szerepel, a másikon az új állapotokat építjük fel. Felesleges a régi rácsra másolni az új állapotokat, inkább a rácsok szerepét cseréljük fel lépésről lépésre. Természetesen nem árt egy gyors nyelv, és mivel igen egyszerű az algoritmus, a legjobb választás a gépi kód.

Problémát okozhat a rács széle, mert odáig nem illik elérnie az alakzatnak. De mivel mi most nem elméleteket igazolunk a programunkkal, nyugodtan fűzzük össze a képernyő alját a tetejével, a bal szélét a jobbal. Egyeseknek biztosan van hozzáférésük párhuzamos gépekhez. Nyugodtan meg lehet osztani a számolást (például feldarabolni a rácsot téglalapokra), de egyrészt vigyázni kell a szinkronra, másrészt a processzorok között folyamatosan ki kell cserélni a szélső sejtek állapotait.

Ne higgyük, hogy ez csak játék: például, ha továbbra is tart a miniatürizáció, nemsokára új módszert kell találni a processzorok készítésére. Nem lehetne ide felhasználni sejtautomatákat? (A kutatások már elkezdődtek!)

Nem öncélú

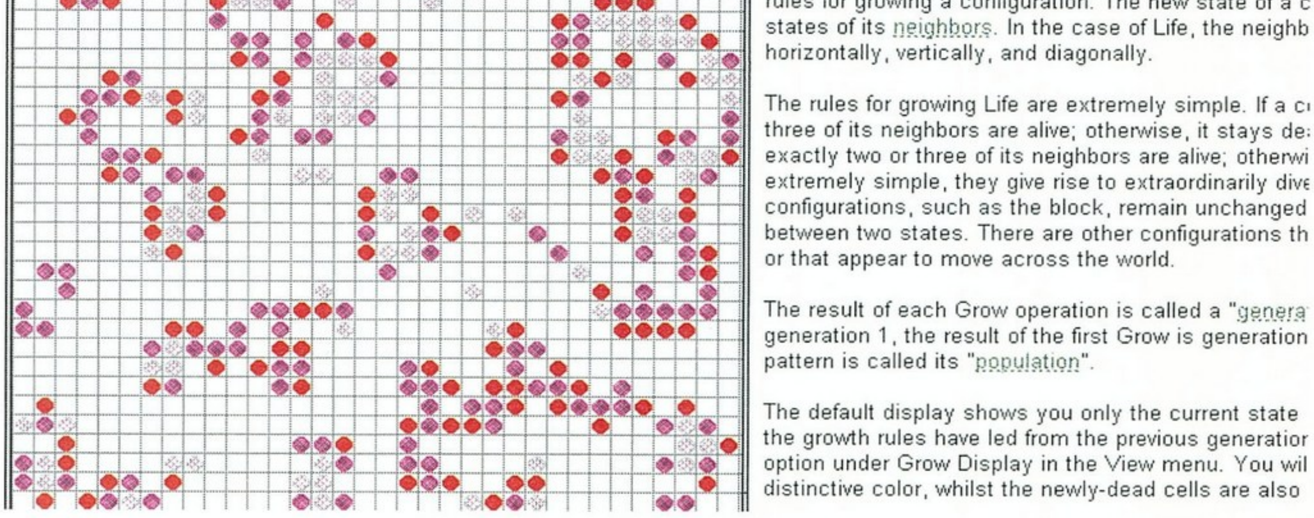
Az eddig felsorolt példák talán meggyőzték az olvasót, hogy a sejtautomatákkal igen jól modellezhetjük a természet jó néhány folyamatát. Hasonló derült ki már régebben a genetikai algoritmusokról és a neuronhálókról. Ezért is kapta az ezekkel foglalkozó tudományág a mesterséges intelligencián belül a „mesterséges élet" elnevezést. Míg a mesterséges intelligencia főleg a számítástechnikai szakemberek által kutatott tudomány, a mesterséges élettel a biológusok, közgazdászok, fizikusok és vegyészek foglalkoznak.

Ha megnézzük a biológia tudományát, akkor azt látjuk, hogy már évszázadok óta csupán a szénalapú élőlényekkel foglalkozik. Mint tudjuk, egy példából messzemenő és általános következtetéseket nem érdemes levonni. (Az élőlényeket csak úgy vizsgálták, ahogy környezetünkben élnek.) De ha itt a Földön csak szénalapú élőlény van (és az ufók nem hagyják magukat megvizsgálni), nincs más lehetőség, mint számítógépes szimulációval azt a mesterséges életet vizsgálni, amely az embertől ered.

A kémiában sem öncélú játszadozás olyan vegyületeket előállítani, amelyek a természetben nem fordulnak elő, hiszen napjainkban ezeknek már fontos szerepük van az iparban is, és hasonlóképpen létrejöhetnek a mesterséges élet ipari alkalmazásai is. (Hogy ez nemcsak álom, hanem élő gyakorlat, és hogy ennek megvannak a veszélyei, arra megpróbálta mindenkinek a figyelmét felhívni a Jurassic park című könyv és film.)

Akit érdekel a mesterséges élet, kapcsolódjon be a comp.ai.alife NetNews csoportba. Némi Internet-kutakodással könnyedén a mesterséges élettel kapcsolatos ftp-helyek, programok, bibliográfiák találhatóak, innen is elindulhatunk hódító utunkra, de ne feledkezzünk el a már megszokott cs.cmu.edu címről sem, mert ott is bőséges a programválaszték.

Aszalós László



Egyszerű szabályok bonyolult következményekkel

Az életjáték is lehet szenvedély

A 70-es évek elején az életjáték valósággal a programozók „szenvedélybetegségévé” vált, és a Time becslése szerint több millió dolláros anyagi kárt okozott a mamutgépeket üzemeltető biztosítótársaságoknak és bankoknak — elvesztegetett gépidő formájában. Sok kutatóintézetben pedig gyakorlatilag megbénult az élet, mert mindenki a komputermonitoron egymást követő „generációkat” figyelte. Egyesek szerint az életjáték ártalmasabb volt bármilyen ismert vírusnál, mások szerint hozzásegített minket ahhoz, hogy jobban megértsük a minket körülvevő valóságot. (Valóban gigászi komputer-e a Világegyetem?)

Az életjáték (a LIFE) John Horton Conway cambridge-i matematikus nevéhez fűződik, aki 1968-ban gondolt először arra, hogy minden bizonnyal lehetne annál a Neumann-féle önreprodukáló automatánál lényegesen egyszerűbbet is tervezni, amelyben nem kevesebb, mint kétszázezer „sejt” van, és mindegyik huszonkilenc lehetséges állapottal bír. Conway valami olyasmire gondolt, amit egy sakk- vagy go-táblán is le lehet játszani, de egy idő múlva olyan bonyolulttá válik, hogy a létrejövő alakzatokat már élőnek kell tekinteni. Ez valójában nem is annyira meghökkentő gondolat, mint amilyennek elsőszörre tűnik, hiszen a Földön is így, lépésről lépésre, az egészen egyszerű struktúrákból kiindulva jött létre az élő sejt.

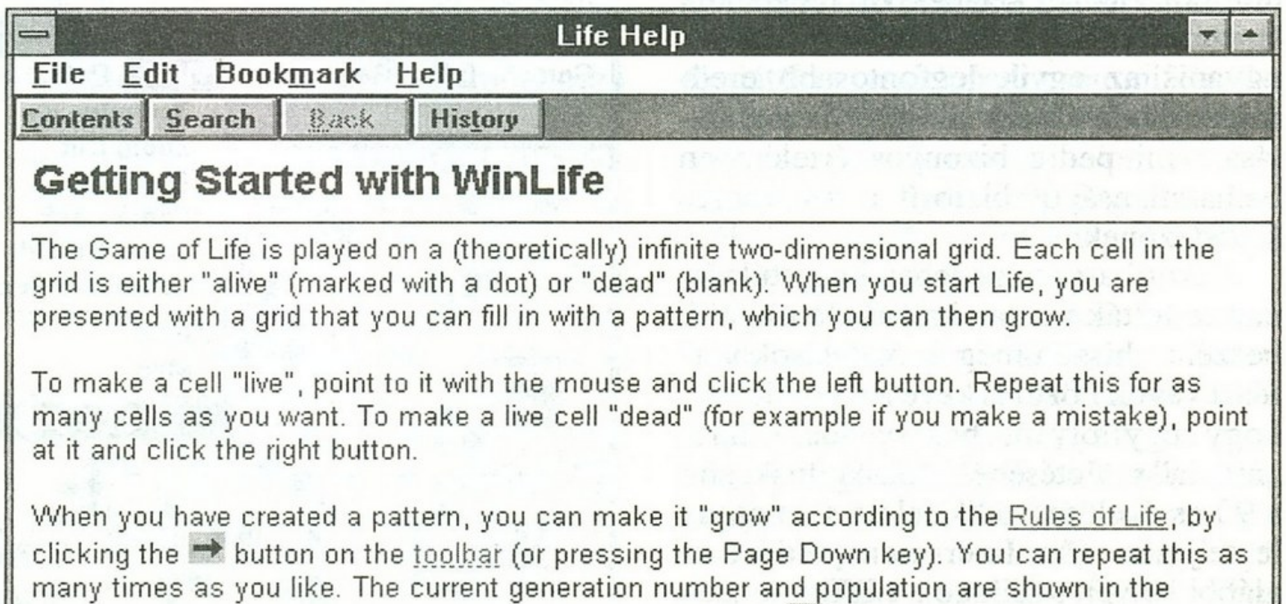
Ahhoz, hogy végül egy ilyen „önreprodukáló teremtmény” jelenhessen meg a színen, nagyon körültekintően kellett megválasztani a „születést”, „túlélést” és „halált” befolyásoló szabályokat, és céljának eléréséhez Conway radikálisan átalakította Neumann huszonkilenc állapotú sejtautomatáját: végül már csak két állapot maradt. Eszerint vagy létezik egy sejt, vagy nem: vagy élő, vagy halott — ennél kevésbé komplikáltat valóban nem lehet elképzelni. Eredetileg felmerült ugyan, hogy esetleg szexuális nemmel is el kellene látni a sejteket, de az csak fölöslegesen bonyolította volna a dolgot: amint hamarosan kiderült, valóban semmi szükség nem volt rá, hogy ezzel is növeljék a játék komplexitását.

Idézzük a klasszikus leírást:

„Az életjátékot egy képzeletbeli négyzethálós táblán játsszák. A négyzetrács alapegységének neve: sejt. Minden sejtnek nyolc lehetséges szomszédja van, azok, amelyekkel érintkezhetnek az oldalai vagy a sarkai. Ha egy sejt a négyzethálós táblán életben van, akkor túl fogja élni a következő lépést (generációt), amennyiben két vagy három élő szomszéddal rendelkezik. Meghal a túlzsúfoltság miatt, ha háromnál több élő szomszédja van, és halálra fagy, ha kettőnél kevesebb élő szomszéd veszi körül. Ha a négyzethálós tábla egyik sejtje halott, akkor a következő generáció alatt is halott lesz, hacsak nem él pontosan három a nyolc szomszédjából. Ekkor a következő generációban megszületik a 'sejt'.”

E szabályok alapján kezdték a LIFE-ot játszani a University of Cambridge matematika tanszékének társalgójában. A játék számára hamarosan kicsinek bizonyult a kávézóasztal, és — leteregetett négyzethálós papírlapok formájában — fokozatosan mind nagyobb területet foglalt el a padlóból is. A lelkes tudósok hónapokon keresztül figyelték a különböző „életformák” létrejöttét — egyesek a periodikusan oszcilláló alakzatok kiszűrésére specializálódtak, mások az egyes formák fejlődését követték figyelemmel (és ez a tevékenység igencsak hasznosnak bizonyult, mivel nemigen lehet előre megjósolni, hogy egy kombináció mikor fog stabilizálódni vagy eltűnni), és éppen ebben a „fejlődésfigyelő” csoportban tevékenykedett Conway kollégája, Richard Guy is, aki felfedezte a később alapvető fontosságúnak bizonyuló siklót (eredeti nevén: glider-t). Ez a különös képződmény minden ötödik generációban felölti eredeti alakját, miközben lassan, de biztosan araszol előre egy adott irányba a négyzethálós táblán.

Mindeddig az életjáték többé-kevésbé a szimpla szabályok nyomán létre-


Life Help

File Edit Bookmark Help

Contents Search Back History

Getting Started with WinLife

The Game of Life is played on a (theoretically) infinite two-dimensional grid. Each cell in the grid is either "alive" (marked with a dot) or "dead" (blank). When you start Life, you are presented with a grid that you can fill in with a pattern, which you can then grow.

To make a cell "live", point to it with the mouse and click the left button. Repeat this for as many cells as you want. To make a live cell "dead" (for example if you make a mistake), point at it and click the right button.

When you have created a pattern, you can make it "grow" according to the Rules of Life, by clicking the button on the toolbar (or pressing the Page Down key). You can repeat this as many times as you like. The current generation number and population are shown in the Status Bar at the bottom of the screen. If you click on the button, the game will grow until it has finished or until you click again on the button.

To start a new game, click on the button on the toolbar. If you want to change the way you started the present game, click on "Copy Start" in the Game menu. This will start a new game whose initial contents are the same as the start of the current game. You can edit this game before starting, using the mouse in the same way as when you create a game from scratch.

jövő végtelen változatosságtól megbűvölt matematikusok magánügye volt. Ekkor azonban Conway arra gondolt, hogy találni kellene egy olyan formációt, amely örökké növekszik. És ez valóban fontos célkitűzés volt, hiszen másként fogalmazva azt mondhatjuk, hogy ha volna egy olyan alakzat, egy „siklóágyú" (glider gun), amely szabályos időközönként egy-egy siklót lőne ki, akkor tulajdonképpen „számítógéppé" is alakíthatnák az életjátékot: az „igen"-nek, vagyis az 1-nek a sikló felelne meg, a „nem"-nek pedig a sikló hiánya.

Siklóágyúk, mindenevők, életkomputerek

A feladat persze meglehetősen komplikáltnak bizonyult, úgyhogy Conway 1970 októberében a Scientific American lapjain ajánlott fel 50 dollárt annak, aki meg tudja oldani a problémát. Az életjátékkal is foglalkozó evolúció-biológus, Karl Sigmund szerint még soha senki nem vásárolt ilyen olcsón ilyen tömérdek számítógépidőt, ugyanis R. William Gosper, az MIT Artificial Intelligence Laboratory programozója is olvasta a cikket, és élete következő négy évét számos „fertőzött" társával együtt a LIFE tanulmányozásával töltötte — eközben természetesen rengeteg számítási kapacitást „pazarolva el".

A fiatal hackerek számára természetesnek tűnt, hogy a „papír-ceruza" módszerről áttérjenek a komputerhasználatra, és ha a komputereket üzemeltető cégek Argentínától Anglián át Ausztráliáig mindenütt ráfizettek is a dologra, Gosper kétségkívül megtalálta a számítását: az ő nevéhez fűződik ugyanis az egyik legfontosabb eredmény, a hőn áhított siklóágyú megtalálása, ami pedig bizonyos értelemben halhatatlanságot biztosít a szerencsés felfedezőnek.

Akkoriban persze lehet, hogy túlzásnak tartották volna „halhatatlanságról" beszélni, hiszen még a „kutatásokban" részt vevők közül is kevesen gondolták, hogy egy olyan, hosszú ideig tartó „mánia" születésénél bábáskodnak, ami a 90-es évek második feléig sem cseng le teljesen. (Az Interneten például az alábbi helyen található a LIFE-rajongók számára fenntartott homepage: http://www.cs.jhu.edu/~callahan/lifepage.html.)

Nem célunk itt a futótűzként elharapódzó életjáték-őrület létrejöttének elemzése (elvégre az életjáték varázsával mindenki megismerkedhet maga is, ha kipróbálja a lap mostani lemezmellékletén található programot); témánk szempontjából annál fontosabb viszont, hogy a következő lépésben megjelent az „életjáték-alapú" számítógép elmélete is.

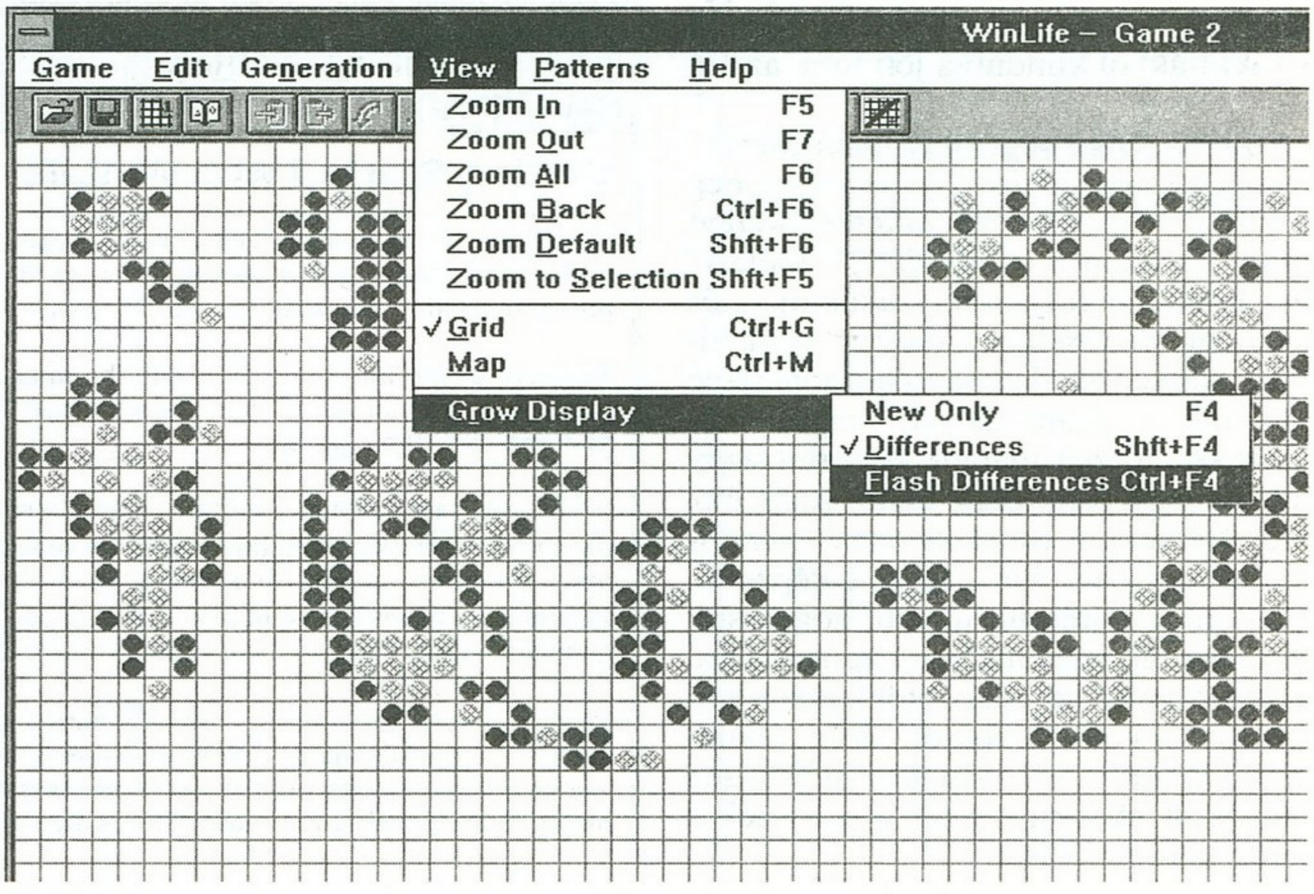
Hamarosan kiderült ugyanis, hogy akinek sikló van a birtokában, az tulajdonképpen mindent meg tud tenni a LIFE univerzumán belül. Ez Gosper szerint annak lehetőségét jelenti, hogy a „tudománycsinálás" Newton, Gauss vagy bárki más követése nélkül valósítható meg: az állattenyésztéstől a rekurzív függvények elméletéig.

Ennek a korlátlan szabadságot kínáló, csodálatos gazdagságnak a megteremtéséhez szükség volt persze más elemekre is: például az úgynevezett mindenevőre (eater), amely a megfelelő irányból beleütköző siklókat megsemmisíti, majd néhány „böffenés" után visszanyeri eredeti alakját; valamint egy igazán nem bonyolult alakzatra, a négy, egymás melletti sejtből álló blokkra (block). És be kellett vezetni néhány „elemi műveletet" is — mint amilyen a siklók ütközése úgy, hogy elpusztítsák egymást, vagy úgy, hogy csak egyikük tűnjön el, mialatt a másik egy négyzettel elcsúszva indul visszafelé. Egy megfelelően irányított sikló elsöpörheti a föld (pontosabban a virtuális játéktábla) színéről az útjában álló blokkot; két sikló meghatározott irányokból érkezve átlósan három négyzettel közelebb, harminc (!) pedig megint csak átlósan három négyzettel távolabb képes tolni ugyanezt a blokkot — illetve megfelelően ütköztetve őket nem csupán mindenevőt és blokkot, de tizenhárom sikló segítségével siklóágyút is létre tudunk hozni.

„Életkomputer"

Innen kezdve már „csupán" Conway ügyességére van szükség ahhoz, hogy létrehozzuk az „életkomputert", illetve végül majd az önreprodukáló életautomatát. Az elemek és elemi műveletek felhasználásával például kanyarodásra kényszeríthetjük a siklókat — három siklóáramlat, illetve ennek megfelelően három programozott siklóágyú segítségével (vagyis egyáltalán nem egyszerű ám a dolog). Az viszont könnyen belátható, hogy ha egyszer a siklók kanyarodásra kényszeríthetők, akkor zárt hurok is alkotható belőlük, hogy ez legyen a számítógép belső memóriája (amelyben persze a siklók megléte vagy hiánya jelenti az egyesek és nullák egymást követő sorozatát).

Az ún. külső memóriát (a tulajdonképpeni winchestert) a blokkoknak az univerzális komputertől mért távolsága határozza meg. A távolságadatokat megint csak siklók megfelelően irányított kilövésével lehet kiolvasni, és nem kell hozzá nagy fantázia, hogy hamarosan rádöbbenjünk: roppant nehézkes, bonyodalmas és lassú lehet egy ilyen gép működése. Ügyelni kell rá, hogy a megfelelő helyen lévő mindenevők elkapdossák és felzabálják a fölöslegessé vált siklókat (akár egyetlennek az elszabadulása is teljes káoszt okozhatna), a blokkokat pedig a tárolandó adatoknak megfelelően két vagy harminc siklóból álló flottákkal kell ide-oda tologatni; a siklóáramlatot jól irányzott lövésekkel ritkítani, és közben (megint csak az univerzális építőegységek, a siklók és siklóágyúk segítségével) arról is gondoskodni, hogy a végtelenség felé



vitorlázó siklók a megfelelő ponton elkanyarodjanak vagy visszaverődjenek...

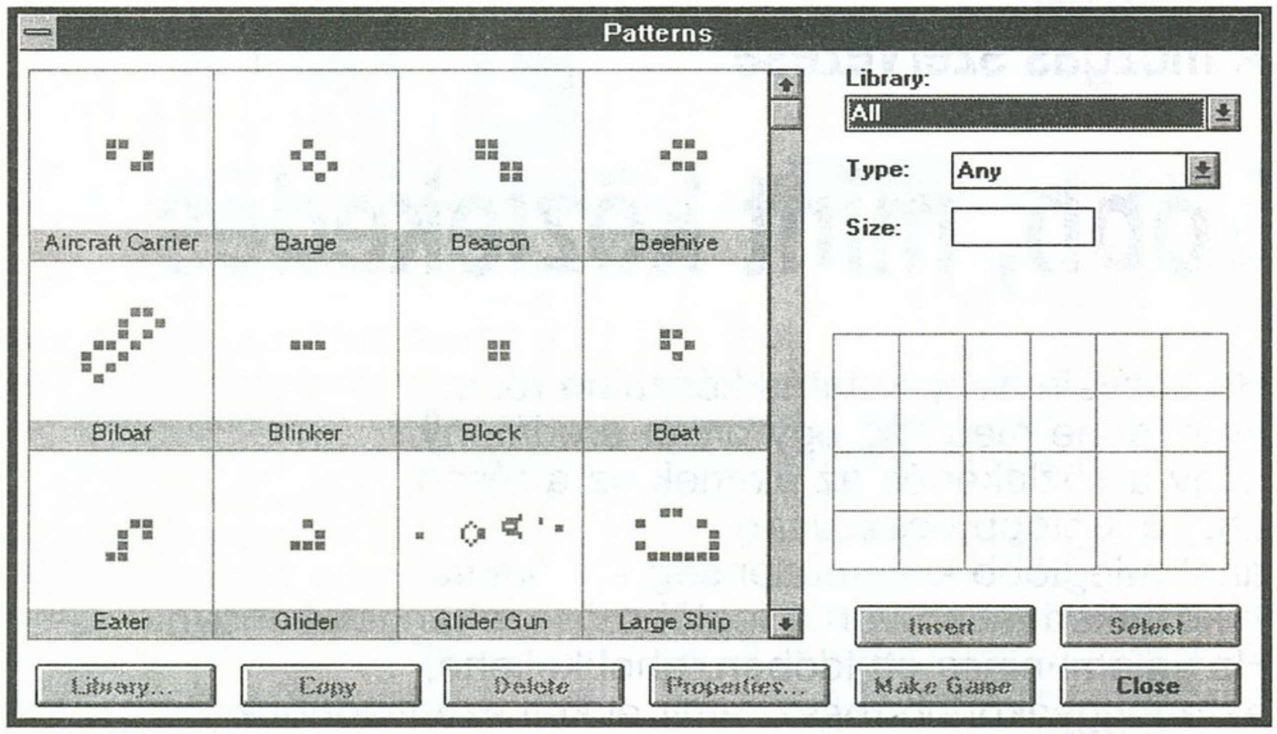
Már az első lépés, a siklóágyúkból, mindenevőkből és blokkokból álló „hardver" megalkotása sem mindennapi teljesítmény: a négy világtáj felől elővitorlázó hatalmas flották megfelelő ütköztetéseivel kell létrehozni az egészet.

A háromdimenziós életjáték és a Taylor-index

Végül persze bonyolultság ide vagy oda, menni fog a dolog, noha egy pillanatig sem lehet kétséges, hogy a fentebb leírtaknál jóval több bütykölésre és ügyeskedésre van szükség a működőképes számítógép létrehozásához (ami a mi fogalmaink szerint ugyan nagyon lassan fog működni, de ez megint másik kérdés). Ám mindenképpen megéri a fáradozás, mert az életjátékból olyan univerzális számítógép alakul ki, amelynek segítségével (miként egyébként akármelyik univerzális számítógép segítségével) akár eddig megoldatlan problémákra is megpróbálhatjuk fellelni a választ. Megvizsgálhatjuk például a híres Goldbach-sejtést (amely szerint minden páros szám két prímszám összege) vagy bármilyen más, nekünk tetsző feladványt, de legalább ilyen szórakoztató, ha továbbra is a „hardverrel" bíbelődünk, és elgondolkodunk azon, hogy ami ilyen kiválóan működik két dimenzióban, miért ne lenne lehetséges három vagy több dimenzióban is.

Carter Bays, a University of South Carolina komputerszakértője az életjáték térbeli megfelelőit választva vizsgálatának tárgyául, különösen a LIFE 4555-öt és a LIFE 5766-ot találta érdekesnek, mivel gazdagságukban és lehetőségeikben ezek hasonlítottak leginkább a Conway-féléhez. Ezeknek a háromdimenziós világoknak akár futólagos bemutatása is nagyon hosszúra nyúlna, úgyhogy itt csak azzal a Bays által kidolgozott nevezéktannal foglalkozunk, amelyben az első két szám az „életszám", ami azt fejezi ki, hogy minimum és maximum hány szomszéd esetén maradhat életben a sejt; a második kettő pedig a „halálszám", mivel az a halott sejt sorsát határozza meg: azt, hogy legkevesebb és legtöbb hány szomszéd esetében születhet egy adott sejt új életre (tehát a Conway-féle játék Bays-száma például 2333 lesz).

Ezt a taxonómiát még az ún. Taylor-indexszel is kiegészíthetjük, amely azt írja le, hogy hány dimenziós életjátékról van szó (az eredeti ezek szerint 2333(2),



Baysé pedig 4555(3), illetve 5766(3) lenne). Ezzel a kezünkben van az univerzális eszköz bármilyen elképzelhető életjáték leírására, de valójában egyáltalán nincs szükségünk holmi 48-as Taylor-indexű szuperalakzatokra ahhoz, hogy szembetaláljuk magunkat a lehető legkomolyabb és legalapvetőbb kérdésekkel: azzal, hogy mi az élet, és mi a minket körülvevő Világegyetem. Vagy legalábbis mi lehetne, ha...

A 0 és az 1 intelligens univerzuma

Nem kíván hosszas bizonygatást, hogy ha lehetséges önreprodukáló automatákat tervezni, akkor azok megépíthetők a kétdimenziós életjáték világában is (elvégre az egész Conway-féle konstrukció felfogható univerzális számítógépként, amint a fentebbiekben már láttuk). Ebből kiindulva elképzelhetünk egy végtelen, négyzethálós síkot, amelyet véletlenszerűen népesítenek be a sejtek, és minden lehetséges formáció előfordul valahol (méghozzá végtelenül sokszor). Lesznek olyanok, amelyek egy (vagy alig néhány) generáció után kihalnak, mások viszont, az önreprodukáló automaták, szaporodni kezdenek, hacsak el nem pusztítja őket egy véletlenül arra tévedő sikló. Biztosra vehetjük, hogy még olyanok is akadnak majd közöttük, amelyek — talán ismét csak egy kósza sikló miatt — nem teljesen pontosan másolják le magukat, és az így létrejövő „lény" életképesebb lesz elődjénél, és így tovább.

Megfelelően hosszú idő elteltével törvényszerűen megjelennek az önreprodukáló automaták kolóniáiból álló, „soksejtű élőlények", és egyes fajok primitív érzékszerveket növeszthetnek maguknak, míg mások támadó- vagy védőfegyvereket hozhatnak létre. És végül az evolúció elvezet majd oda is, hogy egyes életjáték-alakzatok valamikor majd megtanuljanak érezni és gondolkodni (bármit is értsünk ezen).

Ez a több mint meghökkentő következtetés természetesen csak akkor állja meg a helyét, ha a rendelkezésre álló sík, az alakzatok kombinációinak száma és a rendelkezésre álló idő valóban végtelen, vagyis igazi matematikai absztrakcióról van szó. Conway szerint az életjátékon belül a „valódi élet" létrejöttéhez akár az egész világmindenségnél nagyobb go-táblára lehet szükség, ez azonban mit sem változtat a lényegen — állítják az életjáték filozófiájának elkötelezettjei —, azon, hogy a végtelen számú kombinációból értelmes lényeknek KELL születniük, akik elgondolkodnak eredetükön és az őket körülvevő világ törvényein. Esetleg felfedezik a Conway által megalkotott alapszabályokat, és ha maguk is kitalálták már egy kétdimenziós sejtautomata-szuperszámítógépeken futó életjátékot (ennek pedig semmiféle elvi akadálya nincsen), akkor eltűnődhetnek azon is, hogy ők maguk vajon nem egy gigászi számítógép programjának a termékei-e csupán (mint ahogy felőlünk nézve bizony, hogy azok).

Fizikai törvényeink jóval bonyolultabbak ugyan, mint a kétdimenziós életjátéké, ez azonban a siklóágyút felfedező Gosper szerint egyáltalán nem zárja ki legalábbis a lehetőségét annak, hogy mi is csak egy multidimenzionális hacker végtelenül virtuóz programjának termékei vagyunk. És annyiban feltétlenül igaza is van, hogy ezzel a kérdéssel kapcsolatban ugyanúgy találgatásra vagyunk utalva, mint ahogy találgatásra lennének utalva Conway életjátékának feltételezett teremtményei is.

Galántai Zoltán

A mozgás szervezése

Több, mint közlekedés

Ha közvélemény-kutatás készülne róla, nem lenne meglepő egy olyan eredmény, hogy a közlekedés az életnek az a része, ahol a legtöbb bosszúság, ahol a legtöbb kellemetlenség ér minket. A közlekedésben van a legtöbb összehangolatlanság. Ha valami nincs ott időben, ahol kellene, az a leggyakoribb rossz, amit el kell szenvednünk, és ez lényegében mindig közlekedési probléma. Ehhez még csak annyit érdemes hozzátenni, hogy a közlekedés valószínűleg még másban is élen jár: a törvényalkotás mellett ez az a tevékenységkör, amely a saját jelentőségéhez képest a legkevesebb minőségi mutatót használja. Nem azért persze, mintha ezekre nem lenne szükség...

Egyesek (és nem kevesek) szerint nem sok érdekesebb területe van a számítástechnika alkalmazásainak, mint a közlekedés. Emellett azonban más is igaz. Alig van szakterület, amelyben nem lehetne vagy nem kellene közlekedési (helyváltoztatási) modelleket használni.

Ha ezeket a modelleket egyik pillanatról a másikra eltüntetnénk, a modern tudomány megbénulna. Kis túlzással azt is mondhatnánk, hogy a közlekedés az anyagi világ legmélyebb és leglényegesebb életjelensége. Ebbe az izgalmas világba igyekszünk bepillantást nyújtani a következőkben — néhány szemléletes modellen.

Az „állapottér”

Ha a rendszer egyes jellemzőinek pillanatnyi értékeit koordinátáknak fogjuk fel, a rendszer pillanatnyi állapotát egy olyan ponttal reprezentálhatjuk, amelynek koordinátái a rendszerjellemzők pillanatnyi értékei. (Nemcsak ez az egyféle „állapottér” van, és nemcsak ponttal, hanem más alakzattal is reprezentálhatunk rendszereket állapotterekben.) Ha az 1. ábra közlekedőedényének alján kinyitjuk a csapot, a közlekedőedény száraiban kiegyenlítődik a folyadékoszlop magassága. Ez csillapodó lengéssel is megtörténhet. Ha „lassan” zajlik le, akkor az egymás utáni állapotokat az ábra középső részén levő „állapottér” egyenes szakasza szemlélteti.

A közlekedőedény-modellek kiválóan használhatók kiegyenlítődési folyamatok vizsgálatában (is). A kiegyenlítődés pedig a közlekedés, a helyváltoztatás — és a numerikus algoritmusok! — világában az egyik legfontosabb jelenség, amelyről bizony elég keveset tudunk. A téma matematikai elemzése az átlagok elméletéhez vezet... Hogy ez milyen érdekes és egyben milyen nehéz terület, azt egy nagyon egyszerű problémával szemléltetjük. Állapítsa meg az olvasó a kiegyenlítődés utáni folyadékmagasságot olyan esetekben, amikor a közlekedőedények nem egyszerű csövek (2. ábra)! (Siker esetén érdemes kettő helyett több egymással „közlekedő” tartály viselkedésének leírásával is megpróbálkozni.)

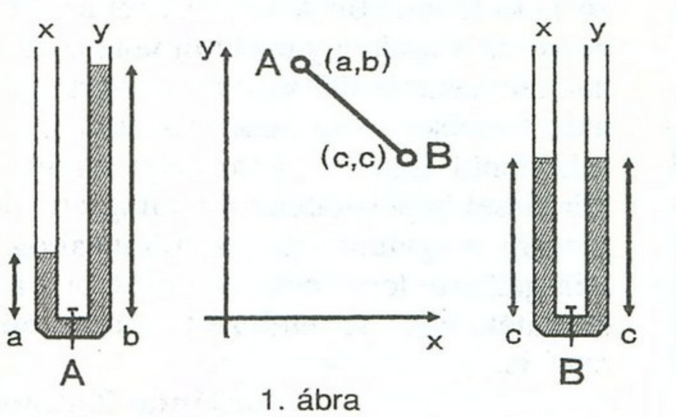
Az állapottér-szemlélet olyan fontos és hatékony eszköz, hogy érdemes begyakorolni. Ajánljuk „agytornának” mindenkinek: ha lát egy görbét, keressen hozzá olyan rendszereket, amelyek működését egy alkalmasan definiált állapottérben modellezni lehet vele! Azzal is mindig közelebb jutunk a jelenségek lényegéhez, ha minden helyzetet, amellyel dolgunk van, igyekszünk minél több mozgási folyamatba beágyazni. (Egész oktatásunkat erre a két alapmódszerre kellene alapozni.)

Mozgástípusok, mozgástörvények

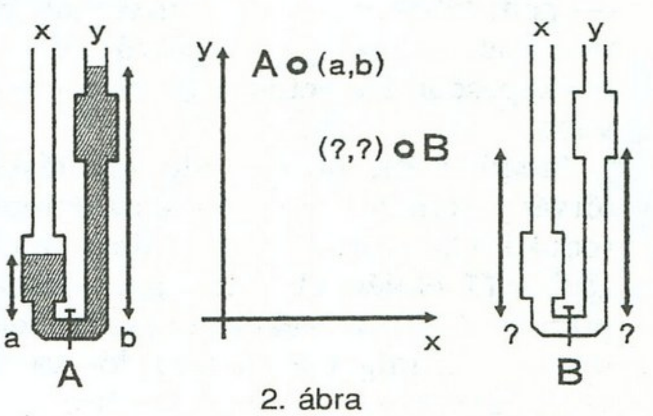
Rengeteg mozgási, változási folyamat van a világban. Ezek összefüggéseinek és törvényeinek tudása maga a tudás. Nem sok mozgástípust, mozgástörvényt ismerünk, ezek alkalmazhatósági köre is erősen korlátozott. A közlekedésben a mozgástípus, a mozgástörvény ismerete azt jelenti, hogy tudjuk, a mozgást végző hogyan végzi a helykiválasztási és a helyelfoglalási műveleteket.

A 3. ábra egy tipikus „véletlennel fertőzött” mozgásfajta összes lehetséges útvonalát szemlélteti. A „Galton-deszka” típusú úthálózaton a fent belépő közlekedő tárgy minden elágazásnál véletlen módon választja ki továbbhaladásának irányát. Ilyenkor az szokott a kérdés lenni, hogy milyen valószínűséggel hol lyukad ki a közlekedő.

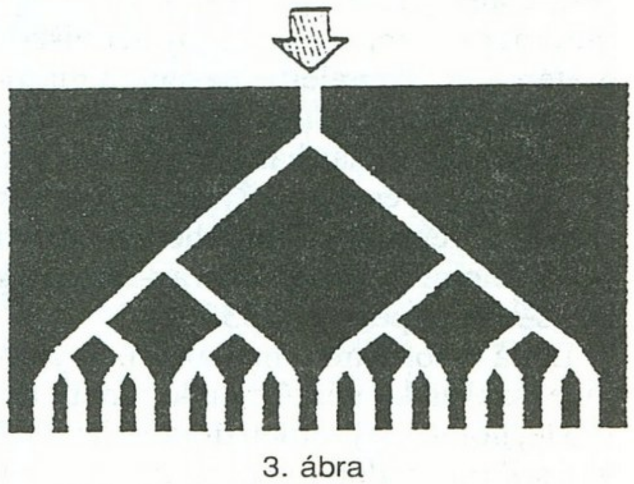
Az adott térrészből adott térrészbe jutás közlekedési alapfeladat. A gyakorlati esetek többségében „változó terepen” kell megoldani e feladatot. A 4. ábra modelljében egy pontszerű tárgynak az A térrészből kell eljutnia a B térrészbe, közben a két térrészt elválasztó „tolózárak” (amelyekben járatok



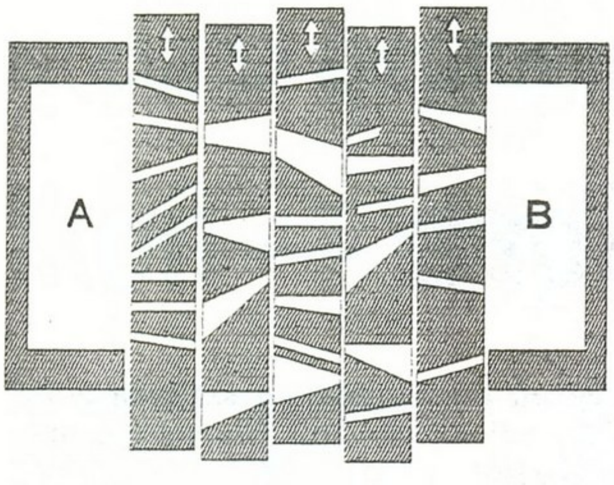
1. ábra



2. ábra



3. ábra



4. ábra

vannak) mozognak. A „járatos tolózárak" által létesített elérhetőségi megfeleltetésektől függ, hogy a feladat mikor megoldható, és mikor nem.

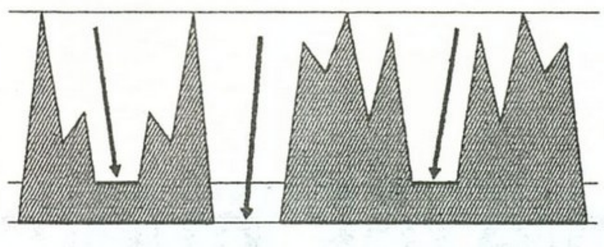
A közlekedésben rengeteg mozgást kell összehangoltan véghezvinni. A klasszikus gépkocsivezetésben például a kormány, a gázpedál, a tengelykapcsoló, a fékpedál mozgatását. Egy konkrét gépkocsi mozgásának mozgástörvényét egyebek mellett ezek között az összehangolt mozgások közti összefüggések határozzák meg. (Sok jó minőségű szimulátor tanúsága szerint e mozgatási folyamatok közti kapcsolatokat már elég jól tudjuk modellezni.)

Több szempontból is nagyon érdekes egy közúti gépkocsisor haladási viselkedése, amely szintén összehangolt mozgásokból tevődik össze. Ajánljuk az olvasónak, készítsen modelleket gépkocsioszlop-mozgásra. Először az egymást követő azonos viselkedésű, azonos gépkocsikból álló kocsisor viselkedését célszerű modellezni. Első lépésben minden kocsi mozgására csak a közvetlenül előtte haladó legyen hatással! (A modell tartalmazzon összefüggéseket két egymás utáni gépkocsi pillanatnyi helyzete, pillanatnyi sebessége és a köztük levő pillanatnyi távolság között!)

Az első kocsi mozgásán változtatva megfigyelhető, hogy vannak olyan mozgásjelenségek, amelyek hatása még az utolsó kocsi mozgásán is kimutatható. Emellett vannak az első kocsi mozgásában olyan jelenségek is, amelyek hatása a kocsisor elején még érzékelhető, de hátrább teljesen elenyészik (kiegyenlítődés!). Harmonikaszerű lengések is kialakulhatnak. (Mikor és miért?)

Kelepcék és vermek

Szinte minden cél elérésére törekvő mozgásnak szembe kell néznie kelepcékkel, illetve vermekkel. Számos ilyen van például a városi közúti közlekedésben is. Az 5. ábra olyan kelepcehelyzetet szemléltet, amelyből összehangolt mozgással kell — ha lehet — kijutni. Az ábrán két forgástest alakú verem van a forgástengelyeken átmenő síkkal elvágva. A vermek alján egy-egy mozgásra képes pontszerű lény (mondjuk hangya) van; nekik olyan összehangolt mozgással kell elhagyniuk a vermeket, és eljutniuk a középső nyíl által mutatott helyre, hogy közben minden pillanatban egyforma magasságban legyenek.



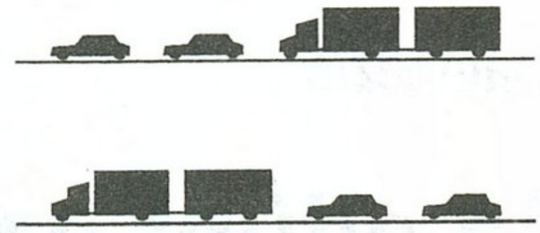
5. ábra

Kintről (is) irányított mozgás

A közúti közlekedés tipikusan olyan mozgás, amelybe „kintről" is beleszólnak — például a közlekedésrendészeti hatóság részéről. Ez a beleszólás a legegyszerűbb esetben közlekedésirányító lámpák működtetésében és jelzések elhelyezésében testesül meg. A nagy probléma — az „optimális" forgalomirányítást megvalósító lámparendszer és jelzésrendszer kialakítása — mind ez ideig még megoldatlan.

A tapasztalat bizonyítja, hogy ha a jelzőlámpák nem automatikusan működnek, hanem azokat egy hozzáértő rendőr irányítja egy olyan helyről, ahonnan kellő áttekintése van a helyzetről, akkor egy csomóponton ugyanannyi idő alatt lényegesen több jármű képes áthaladni.

Az is megfigyelhető, hogy ugyanaz a járműtömeg más sorrendben gyökeresen másképp viselkedik. Érdemes tehát a közlekedésirányítást úgy finomítani, hogy a járművek sorrendjébe is beleszólunk (ez nem lehetetlen!). Mindenki érzi például, hogy egy gyors ciklusú jelzőlámpa előtt nem mindegy, hogy milyen sorrendben állnak a járművek (6. ábra).



6. ábra

Hiányos, ellentmondó és valótlan

A tényleges közlekedésre általánosan jellemző, hogy a fenti alcímmel jellemzett információk alapján zajlik benne a mozgás. Nemcsak azért, mert ellopják az akkumulátort a vasúti kereszteződés jelzőlámpájából. Nemcsak azért, mert közlekedési táblák tömegét szedik le a fémgyűjtők. Nem is csak azért, mert repülőtéri tűznél a hangosbeszélő olyan utat jelöl meg menekülési útként, amely a tűzbe vezet, hanem más okok miatt is. Ezek között az emberi feledékenység, a közönséges tévedés és a hanyagság a leggyakoribb, de a gonosz nemtörődömség és alantas érdek is szerepet játszhat.

(E sorok írójának határozott tiltakozása ellenére például még azután is hosszú ideig csak olyan füzetekkel látták el a kisiskolásokat, amelyeknek hátára az a jó tanács volt nyomtatva, hogy az útra lépés előtt nézzenek balra, az út közepétől pedig nézzenek jobbra, amikor pedig már régen egyirányúsították Budapest utcáinak többségét.)

A tényleges — tehát a hiányos, az ellentmondó és valótlan — információk alapján zajló közlekedésre mind ez ideig semmilyen használható modell nem készült. Ebből következőleg tehát a tényleges közlekedés irányítása — amint ezt naponta tapasztaljuk — nem is lehet túl magas színvonalú.

Mozgás absztrakt terekben

Minden problémamegoldás lényegében közlekedési probléma megoldása: haladás a kezdeti állapotból a megoldás állapotába, akadályokon keresztül, nehézségeket leküzdve. Vannak műveletek, amelyekkel helyünket, helyzetünket változtathatjuk. Ezek „ügyes" alkalmazásával kell célunkat elérnünk vagy megközelítenünk, egy olyan „közegben", amely sokban hasonlít a közönséges „metrikus térre". (Gyakran mondjuk például, hogy „Milyen messze vagyunk a megoldástól?", vagy hogy „Egy lépéssel közelebb jutottunk a megoldáshoz".)

Közismert és széles körben használt kifejezés az „adatforgalom". Egy jó programnak nemcsak az a sajátja, hogy a legfontosabb felhasználói igényeket jó minőségűen, kényelmesen és biztonságosan elégíti ki, hanem az is, hogy ez gyorsan is történik. A sebesség pedig nagyrészt az adatforgalom ügyes megszervezésétől függ.

A programok szolgáltatásainak „útvesztőjében" könnyű eltévedni. De ha nem is tévedünk el, gyakori panasz, hogy sok időt igényel, míg a kívánt szolgáltatást „elérjük". Ne feledjük, emögött is közlekedésszervezési probléma húzódik meg!

Pogány Csaba

Anamnézis

Az eltűnt szervezők nyomában

Szervezettségben, szabályozottságban jelentős különbség van a jól vezetett vállalkozások és a gyengén irányítottak között. A vezetési színvonal különbsége pedig tükröződik az informatikai rendszerek színvonalának különbségében is. De Magyarországon még a jó informatikai átlag is elég alacsony. Vajon mi lehet az oka?

Az informatikai elmaradottság több okra vezethető vissza.

Néhány lényeges és gyakori tényező a következő:

A rendszerek célszerűtlenek, hatékonyságuk megkérdőjelezhető. Az üzleti célokat nem szokás pontosan definiálni, a számszerűsítés pedig szinte mindig elmarad. Vajon hol tűzik ki célul, hogy a készletezési rendszer segítségével a készleteket csökkentsék 20 százalékkal? Az üzleti célok azonosítatlansága nem véletlen. A tervgazdasági lózungok lejáratták a célokat. A célok teljesítésének hiányzó számonkérése pedig értelmetlenné tette a célok megfogalmazását, azt az erőfeszítést, amelyet a többféle cél kompromisszumába fektettek.

Az alkalmazások szervezetlenek. A megvalósítás a jelen és múlt eljárási szabályaira épít. A folyamatok írásbeli utasításai hiányoznak, vagy túl általánosak. Egy új rendszer bevezetésével ésszerűen újra kellene szabályozni az összes folyamatot, és írásbeli utasításokat kellene készíteni minden résztevékenységre. A szervezetlenséggel függ össze, hogy sokszor lassabb az eljárás számítógéppel, mint kézi módon. Ha ez így alakul, akkor az adott rendszer garantáltan rombolja a munkamorált, mert megingatja a hitet a számítástechnikai alkalmazások értelmében és hasznában.

Kevés a folyamatba épített ellenőrzés. Egy rendszer korrekt működését a bele vagy hozzá épített ellenőrzések rendszere teszi lehetővé. A teljesség, zártság és valódiság ellenőrzéséhez vagy magából a folyamatból veszünk adatokat, vagy éppen az ellenőrzéshez képezünk új adatokat (ellenőrző számokat, értékeket).

A rendszerek nem kellően teszteltek. Ma már hibás alapszoftvereket is lehet vásárolni. Az alkalmazási rendszerekről e szempontból ne is beszéljünk! A „teszteletlenségnek" két fő indoka van: (1) Az alapos kipróbálás hatalmas költségekkel jár, amit sok termék nem visel el akkor, ha a hagyományos módszerekkel készítjük. (2) Az öldöklő verseny arra kényszeríti a fejlesztőket, hogy idő előtt lépjenek piacra a rendszerrel.

Külön kell szólni arról, hogy a nagy online rendszerek elvileg is tesztelhetetlenek. Ezeknél a sok felhasználó olyan informatikai helyzeteket produkál, amelyek a szokásos tesztelési eljárásokkal nem állíthatók elő. Az élő működés tesztelése csak részben lehetséges, így a kritikus helyzetek ellenőrizhetetlenek. (Például a rendszer nappal működik, tesztelni pedig éjjel lehet, tartalék számítógépen kell a próbát elvégezni stb.) Még egy hatalmas nehézség van a nagy rendszerek tesztelésénél. Kritikus helyzetek általában csak a rendszer teljesítményhatárainak megközelítésekor alakulnak ki. Márpedig a teljesítményhatárt elérni nehéz, és nem is veszélytelen.

A szervezési problémák tovább sorolhatók lennének. A további boncolgatás helyett most legyen elég annyi, hogy már a kiindulást jól kell megoldani: nagyon pontosan megfogalmazott üzleti célokból kell kiindulni. A meghatározott üzleti célokat szinte a lehetőségek határain túl is számszerűsíteni kell. Például nem elég annyit kitűzni, hogy csökkentsük az ügyfeleknek történő válaszadás idejét, hanem hozzá kell tenni, hogy például két hétről három napra kell leszorítani azt. Vagy másik esetben egy szálloda vezetőjének a hajnalig nyitva tartó bár bezárása után 1 órával legyen pontos adata a szálloda előző napi teljesítményéről, bevételéről, nyereségéről.

Az üzleti célokat jobbára nyereséghez és időhöz szokták kötni. Ez nem elég. Nagyon fontos, hogy minőségi és adatminőségi elvárások is legyenek! Például ki kell tűzni, hogy bemenő adattípusonként az adatok minősége milyen megbízhatósági szintet érjen el. E mutatónál már a kiindulási mérték meghatározása is gondot okzhat (nem ismerjük pontosan azt, hogy a mai adataink mennyire pontosak). Ez a nehézség azonban nem ok arra, hogy az elvárt jövőbeni szintet ne tűzzük ki, azt a szintet, amelyet éppen a rendszer használatával kívánunk elérni.

Egyszerre több üzleti célt is ki kell tűzni, lehetőleg történjen meg minden alapvető cél megfogalmazása. Ezek között prioritásokat kell megállapítani, mert a gyakorlatban léteznek egymásnak ellentmondó üzleti célok. A célok prioritási sorrendje szabja meg a rendszer készítésének vagy kiválasztásának kereteit.

A jó, célszerű informatika kovászként segítheti a teljes szervezet harmonikus és célszerű működését, valamint élen járhat a megújulásban. Az informatika bizony szolgáltatás és következmény. Az informatikus munkájára azonban ez a sommás megállapítás nem lehet felmentő! Ez az alapvető oka annak, hogy nem mindegy, miként teszi a dolgát az, aki erre a „foglalkozásra" adja a fejét. Ezért az informatikus felelőssége igencsak nagy.

A nagy felelősségnek ellentmond az, hogy — sajnos, még napjainkban is, hát még egy-két évtizede... — informatikus, szervező gyakorlatilag bárkiből lehet. Ha némi programozási képzésen sikerült túljutnia, vagy ügyesen a szervezési egységhez került, máris az adott hely „szakembere" lesz. Holott a programozási vagy windowsos tudás még nem jelenti azt, hogy a problémákat is azonosítani tudja, hogy megoldási alternatívákat képes felállítani, és jó megoldásokat ad bizonyos feladatokra. Az érzéken és tehetségen túl valóban sokat kell tanulni és gyakorolni, míg valaki mester lehet e szakmában.

Homonnay Gábor

ICOCUT 1.1 & ICOTOBIN 1.1

Ikon — egy képből kivágva

Az ikonok használata, manipulálása a Windowsra írt programok esetében sem mindig egyszerű. Ugyanakkor gyakran nehézkesnek is tűnik a pixelenkénti képszerkesztés. Sokkal egyszerűbb megoldás lehet esetleg kész képekből kivágni az ikonnak valót. Ezt végezhetjük el az ICOCUT programmal.

A program első esetben paraméterként a képfájl elérési útvonalán (path) indítható (ha nem adtuk meg, a program bekéri), és a hagyományos 32×32 pixeles ikonok kivágására alkalmas. Azonban 'B' kapcsolóval a nagyobb, 64×64 pixeles ikonforma előállítása is lehetséges, amelyet szintén felhasználhatunk az „Ablakban". Az első ikont mindig a 16-színű .BMP fájl nevével menti el, de .ICO kiterjesztéssel. Lehetőség van azonban az utoljára feldolgozott képből új néven további ikonok kimetszésére is. A méretet menet közben csakis az új kép behívása előtt egérrel, az ikongombon, illetve a 'B' betűvel a billentyűzetről lehet módosítani. Ezért megoldható, hogy az ikon kivágása helyett az Escape billentyűvel a zárómenühöz jussunk, majd beállítva az új méretet, új ikon kivágását kezdeményezzük. Az ekkor megjelenő inputsorról ismét az Escape billentyűvel továbblépve, az előző képhez jutunk vissza.

A program kezelni tudja a Windows .BMP és .ICO fájlokban szokásos, 16-színű két leggyakoribb palettát. Ikonok kivágására más 16-színű palettát választó esetekben is lehetőség van. De ez később problémát jelenthet (például az ICOTOBIN mellett kapott fájlok felhasználásakor DOS-os programokban). Alapesetben, referenciaként az először megjelenő egérkurzor színe feketén keretezett fehér. A párbeszédpanelek alapszíne világosszürke, fekete + világospiros betűkkel a gombokon. A címmező sötétkék alapon sárga feliratos. A program rendszerigénye 286-os vagy nagyobb teljesítményű processzor. Támogatja a DOS-EMM386 használatát is (az utoljára feldolgozott kép tárolása alapvetően ennek felhasználásával, az EMS-ben történik meg). EMM vagy más kompatibilis memóriamenedzser nélkül, ha lehet, szintén tárolja az utolsó képet, s amennyiben ez sikertelen lenne, akkor ismételten a lemezről olvassa be a .BMP-fájlt, ami lassítja az ismételt kirajzolást. Amennyiben a program memóriaütközést okozna, a memóriahasználat minimálisra csökkenthető a programot 'M–' kapcsolóval indítva. Ekkor a kép soha nem tárolódik a memóriában.

Az ICOCUT program használatához egér szükséges. Az egérkurzor mozgástere a kivágandó ikon méretéhez igazodva a kép határain belül korlátozódik. A feldolgozható maximális képméret 640x480 pixel.

Amennyiben a kivágott ikonokat, illetve a Windows 16-színű .ICO állományait DOS programokban kívánjuk felhasználni, az ICOTOBIN programmal konvertálhatjuk a programba beolvasható, illetve „linkelhető" bináris formátummá, illetve Turbo Pascal programokban felhasználható BYTE konstanstömbbé. Alapesetben a program mindkettőt létrehozza, de parancssorból a '/B' kapcsolóval kérhető csak a bináris állomány, illetve a '/C' kapcsolóval csak a konstanstömb létrehozása. '/C+' kapcsolóval a Pascal-típusú BYTE konstanstömb helyett a C nyelvű programokban felhasználható CHAR konstanstömb előállítására is lehetőség van. A konstanstömböt a program '.BYT' kiterjesztéssel írja. (Mindkét típust!)

Az átkonvertált ikon megjelenését követően a rendszer pillanatnyi beállításáról kapcsolóként funkcionáló gombok tájékoztatnak a képernyő jobb oldalán. Kivilágított állapotban az elvégzendő konverzióknak megfelelő gombokat mutatja a program. Amennyiben valamennyi beállítást kiiktatnánk, a program az alapértelmezett beállításba vált át (bináris állomány + Pascal BYTE-tömb).

A konvertáláskor kapott bináris állományokat programíráskor a bináris fájlból egy pufferterületre olvastathatjuk, amelynek a címe megadható a PUTIMAGE eljárásnak. Azonban más lehetőség is van a felhasználásra, mely lehetővé teszi, hogy programunk végső alakban (.EXE-ként) ne igényeljen külső forrást. Ekkor azonban legyünk tekintettel a konstanstömbök nagyobb memóriaigényére a DOS-os alapmemória-területen.

A TURBO PASCAL programokhoz a későbbiekben a .BIN fájl a BINOBJ.EXE segítségével .OBJ fájlba is konvertálható, és a programba linkelhető. Ebben az esetben a betű- és driverkészletekhez hasonlóan a „regisztrálást (bekötést)" végző eljárás címét kell megadni a PUTIMAGE eljárásnak (1.). A konstanstömb esetében azt be kell másolni a program deklarációs részébe, majd a tömb memóriacímét kell átadni a PUTIMAGE eljárásnak (2). Utóbbi megoldást a C nyelvben is alkalmazhatjuk (3):

```
1;)

{* TURBO
PASCAL-hoz*************************************** *}

{$M $4000,$ffff,655360}

{A HEAP-ben min. 64 K-t kérünk, és „ráülünk" a
memóriára, bejelentve a lehetséges maximum igénylését}

{$F+} {Biztosítjuk a távoli hívást}

{A procedúra csak az .OBJ bekötését végzi, és az
eljárás memóriacímén fogjuk elérni az ikon bináris
adatait az .EXE-ben}

procedure REGICO;external; {$L MYICO.OBJ }

VAR

Imager:pointer;

{Inicializáljuk a grafikus rendszert (InitGraph)}
```

```
Imager:=@REGICO;
{A pointert állítjuk rá a korábbi procedúra címére,
mivel a PutImage nem fogad el eljárást paraméterként}
PutImage(0,0,Imager^,NormalPut);

2;)
{* TURBO
PASCAL-hoz************************************ *}
{$M $4000,$ffff,655360}
{A HEAP-ben min 64 K-t kérünk; 1. előbb}
{$I MYICO.byt}
{A fordítóval beépíttetjük a kapott konstanstömböt
tartalmazó fájlt}
{Inicializáljuk a grafikus rendszert (InitGraph)}
{A konstanstömb memóriacímét átadjuk a PutImage
eljárásnak}
PutImage(0,0,@MYICO,NormalPut);

3;)
/* TURBO
C-hez*************************************************/
#include ________.h
#include graphics.h
# include „myico.byt" /* Transzformalt ico-fájl */
void main( )
{
  int driver = VGA;
  int mode = VGAHI;
  initgraph(&driver, &mode, „d:\\tc\\bgi"); /*
Grafikus konyvtar */
  setfillstyle(7,7);
  bar(0,0,getmaxx( ),getmaxy( ));
  putimage(0,0,Myico,COPY_PUT);
/* Myico helyett a char-tomb */
  getch( );
  closegraph( );
}
```

Bár csak a teljes programot leíró C rutinban van jelezve, az itt említett két program legalább VGA monitoron fut megfelelően, illetve az ikonok csak e monitor magas felbontású üzemmódjában adnak megfelelő képet a monitoron.

Kívánság szerint lehetőség van az ikonnak .BMP kiterjesztésű, 16 színt használó képfájlba való konvertálására is. Ez a későbbi képszerkesztést is szolgálja, amennyiben az átrajzolt képet az ICOCUT programmal visszakonvertáljuk ikonná. A program e lépésnél csak mérsékelt I/O ellenőrzést végez (például a lemezkapacitást nem, csak a felülírást ellenőrzi!).

Simay Endre István

MagiCub 1.1

Képkocka — kicsit másképp

Ez a kockajáték a korábbi, vagyis a különböző színű sorokat kirakó játék (MagiCub 1.0) egyfajta átirata. Lényege ennek is az, hogy az eredeti látványt a keveréseket követően visszaállítsuk. Azonban itt már nem soronként egyenértékű négyzeteket kell visszasorakoztatni, hanem egy kép részleteit (egy Windows-kompatibilis .BMP képét).

A MagiCub 1.1 alapelve megegyezik az előző változatéval, tehát nem puzzle-szerűen rakjuk ki, hanem egyszerre egész sorokat, illetve oszlopokat mozgatunk, csúsztatunk el. (Mint a bűvös kockánál.) Az eredeti kép a játékok elején, a betöltés során teljes egészében, majd négyzetekre hasítva is megtekinthető, és 3 és 20 között megadható a keverések száma. A gép mozgatja el a sorokat, oszlopokat — véletlenszerűen.

A keverés során ugyanazokat a rutinokat használja, amelyek a játékos által vezérelve később is végrehajtják a képrészletek elmozdítását. A teljesen véletlenszerű elrendezéssel ellentétben ez biztosítja, hogy elvben minden keverést követően kirakható a kép. De a gyakorlatban... Mindenesetre először ne húszas keverési szinttel kezdjünk, és türelemmel viseltessünk gépünk iránt az ötvenedik lépés után is.

A játék kezelése azonos a korábbi verzióval, de itt is megismételjük:

A játék vezérlése egérrel a legcélszerűbb, bár billentyűkkel is megy: a kis színes négyzetek elmozdulásának irányát víszintesen a q és r, függőlegesen a v és m billentyűkkel lehet vezérelni. A kisbetűs forma balról jobbra, illetve fentről lefelé mozdítja el a sorokat, illetve oszlopokat. Ellenkező irányba a SHIFT és a megfelelő gomb lenyomásával tudunk mozogni.

Amennyiben egér nincs installálva a gépen, a megfelelő betűk jelennek meg a nyilak helyén. De menet közben is kérhető segítség az F' gomb lenyomásával, illetve az egér érvényesítésével a ? ikonon. Játék közben az ESC billentyűvel, illetve az Alt+F! billentyűkkel léphetünk ki, 'U'-val kérhető új játék, illetve a 'H' billentyűvel lehet a hangot letiltani/engedélyezni. Egérrel a megfelelő ikonok is használhatók.

A felhasználható kép bármely 16-színű Windows-kompatibilis .BMP fájl lehet. Azonban a mérete korlátozott alulról is (ki kell férnie a gomboknak) és felülről is (el kell férnie a képernyő alsó részén). Az alapértelmezett képtől (CASTLE.BMP) eltérő képet a parancssorból adhatjuk meg: a teljes elérési úttal. Ha ennek paraméterei nem megfelelőek, vagy pedig a képfájllal van problémája a programnak, figyelmeztetés mellett lehetőséget ad a javításra. De minden esetben olyan képet használjunk, amelynek mintázata jó elosztott, és képrészletenként jellegzetes. Ez kirakáskor a megfelelő hely azonosításához igen fontos lehet.

A program EMS-t használ, ha lehet, és legalább VGA monitort igényel. Bár memóriaigénye a képrészletek tárolásának következtében magasabb a korábbi verziónál, memóriaütközést Windows alól indítva sem tapasztaltam, de esetleg előfordulhat.

Simay Endre István

INFORMÁCIÓKÉRÉS: 0705 ▼

4M COMPUTERBONTÓ

„4M" Műszaki és Kereskedelmi Kft.

1072 Budapest, Klauzál u. 32.
Tel.: 26-79-560

Új és használt számítástechnikai berendezések, alkatrészek eladása-vétele, elfekvő és leselejtezett készletek nagy tételben való megvásárlása.

Használt, működő fénymásolók és computerek.

Nyitva hétfőtől péntekig: 10–18 óráig, szombaton: 9–13 óráig

INFORMÁCIÓKÉRÉS: 0729 ▼

DIALCOM 1414

MAGYAR FAXMODEM

magánemberek, kisvállalkozások ideális kommunikációs eszköze



▲

14 400 bps,

GIII FAX,

V.42bis,

MNP5

▼

Magyarországi gyári beállítások!

Teljes magyar nyelvű dokumentáció!

Típusengedély!

SCI-MODEM

Gyártó: SCI-Modem Kft., Tel.: 270-9020

MEGVÁSÁROLHATÓ A SZÁMÍTÁSTECHNIKAI SZAKÜZLETEKBEN!

ELENDER® COMPUTER

MaxLyb 26XT Jukebox



26 GB, T3 Optikai drive, 4 MB Cache, 20 db. lemez, SCSI II., lemezcsere: 2,5 s., MTBF: 500,000 óra, MSBF: 500,000 méret: 46x22x66 cm.

Maxoptix T4 2600

az ideális optikai drive audio és video alkalmazásokhoz



- 2.6 GB
- 30 ms hozzáférési idő
- 4.5 MB/s
- SCSI II.
- 1 MB Cache
- 247x64x264 mm
- 100.000 óra MTBF
- Novell bevizsgált

ELENDER INTERNET

1087 Budapest, Hungária krt. 8. Tel.: 134-5008, 114-0532 Fax: 133-4347
IX. Ferenc krt. 16. Tel./Fax: 218-2858 * XIII. Csángó u. 13. Tel./Fax: 270-3097
Debrecen, Piac u 57. Tel./Fax:(52) 413-795 ■ Szeged, Madách u. 15. Tel./Fax:(62) 310-269 ■ Veszprém, Botev üzletház Tel./Fax:(88) 428-235 ■ Szombathely, Hunyadi u. 45. Tel.Fax:(94) 336-479 ■ Pécs, Klimó Gy. u. 13. Tel./Fax:(72) 312-820 Nyíregyháza, Nyírfa tér 5. Tel./Fax: (42) 405-666 ■ Miskolc, Szent István u. 1. Tel./Fax: (46) 340-860

Nyitva: hétfőtől péntekig 9-17 óráig

INFORMÁCIÓKÉRÉS: 0709 ▼

INFORMÁCIÓKÉRÉS: 0725 ▼

EPSON

NYOMTATÓK és TARTOZÉKOK

IBM

SZÁMÍTÓGÉPEK

TELJES VÁLASZTÉKA

RÉSZLETFIZETÉSRE IS KAPHATÓK

QWERTY High Tech Kft: 1114 Budapest, Bartók Béla út 9. Tel.: 166-5419

A négyezer éves titok nyomában

Rokon vonások a göröggel

Egy megkezdett sorozatnak nagyon nehéz a végére érni. Jóllehet a korong titkainak fejtegetései már távol kerültek a számítástechnika tárgykörétől, úgy érezzük, kötelességünk az elindított gondolatfolyamot az elképzelt kifejletig eljuttatni. A sorozat szerzőjének szándéka szerint ez nem lesz a korong teljes megfejtése (a nyilvánvalóan szalonképtelen részletek miatt), pusztán a kikövetkeztethetőségét ígéri. Annak érdekében viszont, hogy a tömény számítástechnikát igénylők kevésbé érezzék ballasztnak a korong fejtegetését, csak kéthavonta adjuk azt közre.

A phaisztoszi korong jeleiről az ismert analógiák alapján bizonyosra vehetjük, hogy CV (mássalhangzó + magánhangzó) típusú szótagjelek. Ennek fontos következménye, hogy a hangértékével együtt megismert szótagjeleket a rendszer mindjárt kettős rendezésnek megfelelő helyükre irányítja a mássalhangzók és a magánhangzók kétdimenziós mátrixterében.

A jelek behelyettesítését így kétféle vetületben tarthatjuk nyilván: egyrészt a valóságos szövegben, másrészt a jelkészlet kettős rendezésű virtuális „szövegében".

A rokonítható írásrendszerek (lineáris B, ciprusi) példájából kiindulva eléggé kézenfekvő az a következtetés, hogy a korong jelkészletének megoldási kulcsát is egy hasonló, n × 5 méretű mátrixban kell keresnünk, ahol az oszlopok felelnek meg a magánhangzóknak, a sorok pedig a mássalhangzóknak (ez utóbbiak közé sorolva az „üres mássalhangzót", vagyis a mássalhangzó hiányát is). A sorok számát pontosan nem ismerhetjük, de a korongon előforduló különböző jelek száma alapján n értéke nem lehet kisebb 9-nél.

A közvetlen jelhasonlítás alapján kikövetkeztetett jelek között 8 mássalhangzó fordult elő (ismét csak az „üres mássalhangzóval" együtt). Kilencedikként minden valószínűség szerint fel kell vennünk az „r"-t, de a lineáris B alapján bizonyosra vehetjük, hogy az ennek megfelelő RA — RE — RI — RO — RU szimbólumai jelölik a LA — LE — LI — LO — LU szótagokat is. Egyéb mássalhangzóval addig ne egészítsük ki mátrixunkat, míg csak nyomós okunk nem lesz rá.

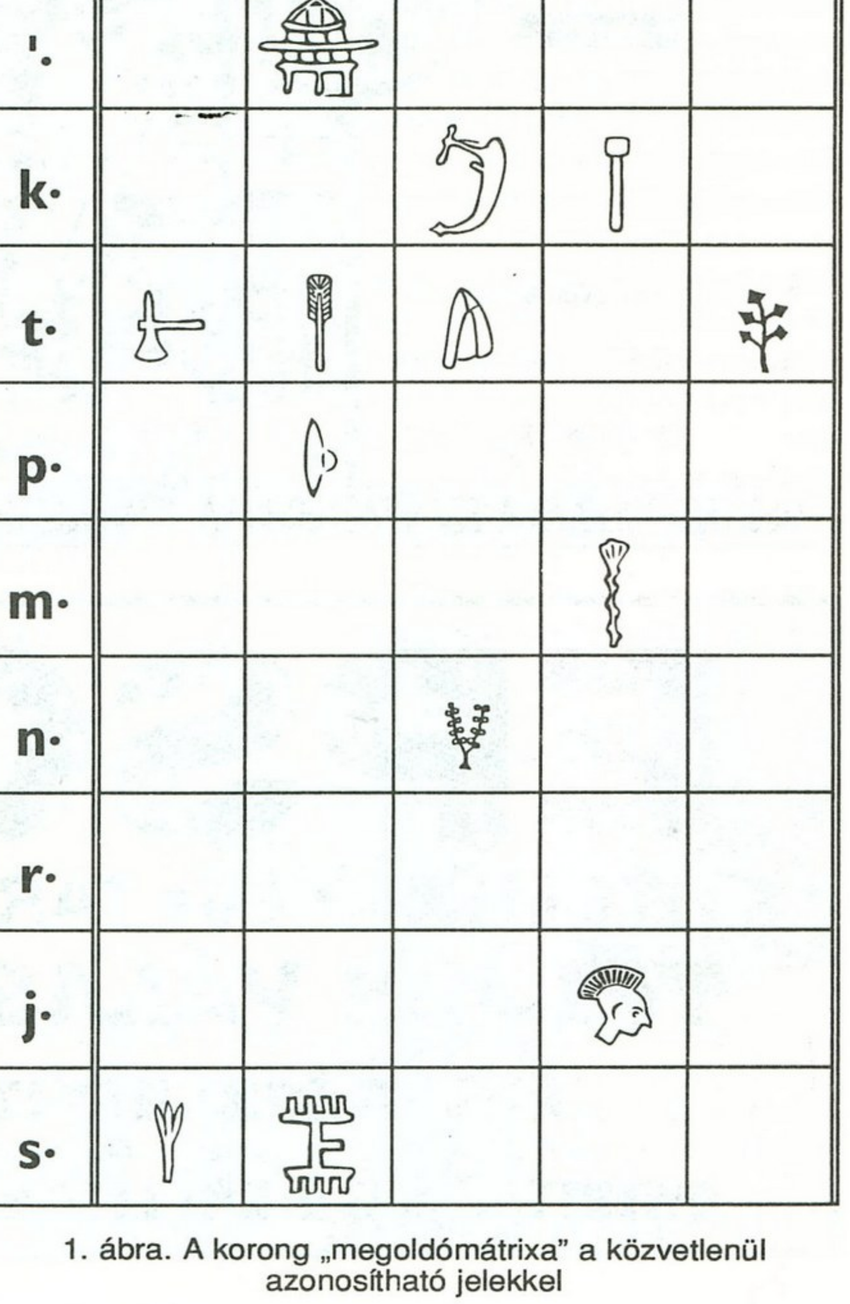
Készítsük el viszont eddigi eredményeink összegzéseképpen a kikövetkeztetett szótagok mátrixát. Az ismeretlen hangértékű jelek helyét üresen hagyhatjuk (1. ábra). Eredménymátrixunkban egyelőre még több a lyuk, mint a jel, meglepő módon azonban van egy sor, amely máris csaknem teljesnek mondható.

Akrobata — akropolis — akrofónia

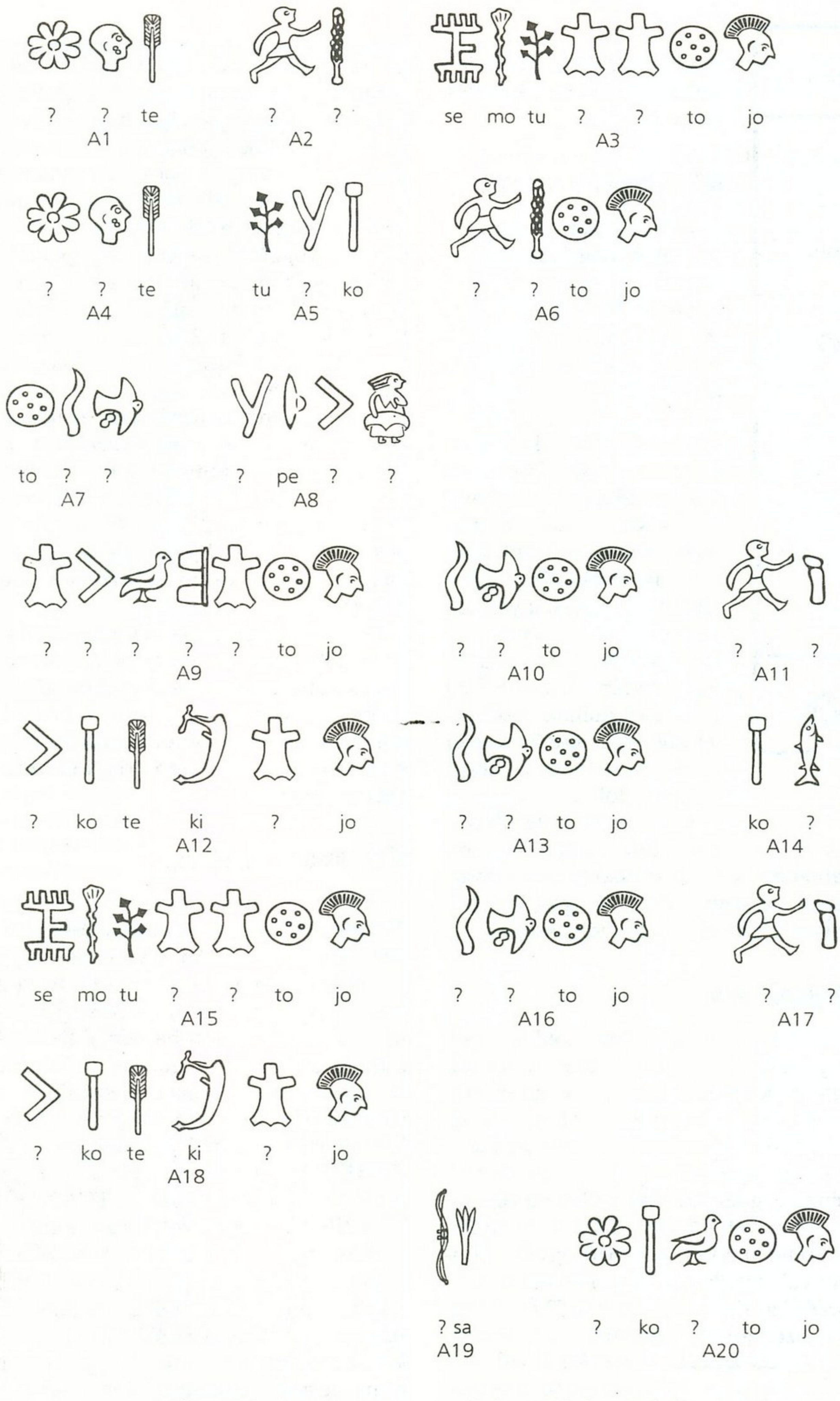
Külön érdemes felhívni a figyelmet arra, hogy eddigi hangmegfeleltetéseink valószínű keletkezésének szinte mindenütt megtalálható volt az ún. akrofóniai indokolása is. (A görög AKRA szó valaminek a hegyét, csúcsát, búbját jelenti, így keletkezett az AKROBATEO 'lábbujjhegyen lépked' szóból előbb a 'kötéltáncos' jelentés, később a még általánosabb AKROBATA; ez a magyarázata a hegycsúcson lévő AKROPOLIS elnevezésének is. Szavakra vonatkoztatva az AKRA a szónak a kezdetét jelöli: minden nyelvben ismeretesek például a szókezdetekből összerakott ún. AKRONYM szavak: IFIP, UNESCO, IFOR stb.)

A korong szótagjeleinek hangtani értelmezése tehát semmiképpen nem tekinthető motiváció nélkülinek. Példaképpen idézzük fel annak a sornak az akrofóniai indokolását, amelyet majdnem teljes egészében sikerült felismernünk a lineáris B jeleivel való összehasonlításból. A TA jele feltehetően a „csákány" képéből (és elnevezéséből) származott (DAbrüsz), a TE jele a „nyíl"-éból (TEkhné), a TI jele pedig jól levezethető a „TIara" képéből és nevéből.

Legnehezebb volt ebből a sorból a korong „ág" jelének olvasatát meghatározni a lineáris B jeleivel való hasonlóság alapján. Tulajdonképpen úgy si-



1. ábra. A korong „megoldómátrixa" a közvetlenül azonosítható jelekkel



2. ábra. Részlet a korong szövegéből egy lépéssel később

került eredményt elérnem, hogy kiválogattam a lineáris B jelei közül azokat, amelyek növényeket jelenthetnek, s így már könnyebb volt összepárosítani ezeket a korong növényábrázolataival. Eredetét tekintve ennek a jelnek TU olvasatához szóba jöhet a) a bakkhuszi áldozati szertartásokon használt THUia (tujafa), b) esetleg a THULLA alakban is említett THALLOS 'fiatal, leveles ág' szó (Hészükhiosz), c) vagy éppen az 'ág', 'új hajtás' jelentésű latin TURIO szó valamilyen rokona.

Hogyan léphetünk tovább?

Akárhogy is van, ritka szerencsésnek mondhatjuk magunkat, hogy a „t" mássalhangzó sora csaknem teljesre sikeredett. A hiányzó TO szótag jelét ugyanis kikövetkeztethetjük előző megfigyeléseink alapján. Emlékezzünk vissza, hogy sok szópárban az „ág" jele számos helyen váltakozik a korongon a „pajzs" jelével (az A7 és A23, A26 és B29, A30 és B21, B18 és B26 szavakban). Feltételezésünk szerint az „ág" — „pajzs" váltakozás mögött a korong készítőjének tudatos törekvését kell látnunk arra, hogy minél több alliterációt vigyen bele a szövegbe, ezzel is fokozva annak varázserejét. Ha ez a feltevésünk helyes, akkor a TU olvasatú „ág" jel alliteráló „pajzs" párjának a TA — TE — TI — TO olvasatú jelek közül kell kikerülnie. Mivel azonban a TA — TE — TI jeleket már ismerjük, a „pajzs" jel megfelelőjének egyedül a TO jel marad.

Valóban beindul a motor?

Ismeretlen írások megfejtésekor az indulás a legnehezebb. Meg kell találni azt a pontot, ahonnan „be lehet törni" a szövegbe, ahonnan a felismert szabályszerűségek azután már segítik egymáshoz illeszteni a részinformációkat. Az egyiptomi hieroglifáknál a „névgyűrűk" (cartouche-ok), az uralkodók neveit kiemelő keretezések segítettek hozzá egyes ismert nevek azonosításához, már a rosette-i kő megtalálása előtt (!). Az óperzsa ékírás megfejtése során arra jöttek rá, hogy bizonyos ismétlődő jelsorozatok helyébe be lehet helyettesíteni ismert szófordulatokat („királyok királya", „valakinek a fia" stb.).

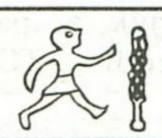
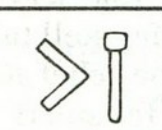
Számunkra — akárcsak a keresztrejtvényfejtésben — a felismert jelek sűrűsödése adhat támpontot a továbbhaladáshoz. A sűrűsödés azonban, mint látható, kétféle vetületben is hasznunkra lehet. Egyrészt segíthet a jelrendszer virtuális „szövegterében", ahol 5-ben van maximálva a jelek száma minden soron belül, minden jel pontosan egyszer fordulhat csak elő, és azonos rendezettség uralkodik a sorokon belül is (a magánhangzók szerint). Másrészt kihasználhatjuk a jelek sűrűsödését magában a valóságos szövegben, ahol előbb a szavakon belül, majd később nagyobb egységeken belül lehet kikövetkeztetni az ismeretlen jelek hangértékét.

A 2. ábrán látható — a szöveg egy része alapján —, hogy hol tartunk most, milyen pontokon érdemes keresni a továbbhaladás lehetőségét.

Jelcsoportok

A phaisztoszi korong faggatása során most érkeztünk el egy kritikus ponthoz. Ha minden igaz, akkor már nem kell megelégednünk egyes jelek olvasatával, hanem aktuálissá válik bizonyos jelek kapcsolódásából származó jelcsoportok vizsgálata is.

Kizárásos alapon sikerült kikövetkeztetnünk a kerek pajzsra emlékeztető

	Előfordulás	%-ban
	**13**	**7,2%**
	6	3,3%
	5	2,8%
	5	2,8%
	4	2,2%
Egyéb	148	81,7%
Összesen	181	100%

3. ábra. Az 5 leggyakoribb szótagpár

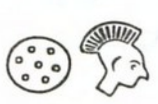
szótagjel TO olvasatát. Gyakoriságban ez a jel — 17 előfordulásával — alig marad el a „harcos" jelétől (19), helyes felismerése ezért különösen fontos. Gondoljunk csak arra, hogy az öt leggyakoribb jel 81 előfordulása teszi ki az összes előfordulásnak (241) kereken az egyharmadát. Amilyen nagy segítséget jelenthet a gyakori jelek helyes meghatározása, annyira félre is viheti az itt elkövetett tévedés a további következtetések láncolatát.

Az egyedi jelek gyakoriságán kívül az előforduló jelpárok rangsora is sokatmondó lehet. A jelpárok gyakorisági listáját egy olyan jelpár vezeti kiugróan magas értékkel, amelynek első tagja a szóban forgó „pajzs" jel (3. ábra). Még beszédesebbé válnak az adatok, ha megfigyeléseink körét leszűkítjük a szóvégződésekre (4. ábra).

Mit mutatnak a számok?

A számszerű vizsgálatok tehát megerősíteni látszanak, hogy e jelpár („pajzs" és „harcos") TOJO olvasatának meghatározásával kezünkbe került a korong szövegének egy rendkívül fontos grammatikai eleme. Hátravan azonban még néhány ellenőrző lépés ahhoz, hogy trófeánkat felmutassuk. Először is igazolnunk kell, hogy nyelvészetileg is értékelhető eredményre jutottunk, másodszor, ha nem is kötelező, de kívánatos feltárni a „pajzs" jel TO olvasatának szókezdetből való eredeztetését (szakmai műszóval: akrofóniai indokolását).

Ha mindez sikerül, akkor megnyugodhatunk, hogy jó úton haladunk, s az elért első komoly eredmény visszamenőleg is igazolja felhasznált hipotéziseink helyességét.

	A oldal	B oldal	A és B
	12 **(38,7%)**	1 (3,3%)	13 **(21,3%)**
Egyéb	19 (61,3%)	29 (96,7%)	48 (78,7%)
Összesen	31 (100%)	30 (100%)	61 (100%)

4. ábra. A leggyakoribb végződés

Meglepő fordulat

Nyelvészetileg elég nyilvánvaló, hogy a -tojo végződés két elemből áll: egy -t- képzőből és egy hozzá csatlakozó -ojo névszói toldalékból. A -t- képző (és számtalan változata) az indoeurópai nyelvek egyik leggyakoribb képzője, gyakran képeznek vele például melléknevet főnévi vagy igei tőből. Hasonló módon, a képző egyik változatával keletkezett a görögben a NAUSZ 'hajó' főnévből a NAUTÉSZ 'hajós' szó (amely mindenki számára ismerős a kozmonauta szóból), a POIEÓ 'alkot' igéből a POIÉTÉSZ 'költő', a POLISZ 'város' szóból a 'városi, polgár' jelentésű POLITÉSZ, majd ebből egy további képzővel a POLITIKOSZ 'közéleti személy'.

Izgalmasabb azonban az -ojo toldalék, amely kimondottan csemege az epikus görög nyelv ismerőinek. Homérosz eposzainak egyik legjellegzetesebb nyelvi sajátsága, hogy a főnevek (és a velük azonos ragozású melléknevek) bizonyos típusában gyakran ez a végződés áll az egyes számú birtokos esetben a későbbi -ou (kiejtésben -u) végződés helyett. Így például 'Priamosz városa' az Iliászban sűrűn szerepel „Priamojo polisz" vagy „polisz Priamojo" alakban, Patroklosz birtokos esete gyakran Patroklojo, Ókeanoszé Ókeanojo stb.

Ennek az archaikus alaknak a nyomai egyébként egyes nyelvjárásokban is elég sokáig fennmaradtak, például Thesszáliában, Boiotiában és Leszbosz szigetén — hogy a szanszkritról ne is beszéljünk, ahol éppen az ezzel az alakkal analóg -oszjo végződés a szabályszerű.

A korong nyelvéről tehát beigazolódni látszik, hogy lényeges pontokon rokon vonásokat mutat az epikus görög nyelvvel. Mindez egy olyan korból, amikor eddigi ismereteink szerint még meg sem jelentek a görögök a történelem színpadán...

Működik-e az akrofónia?

Adósak vagyunk még a „pajzs" jel TO olvasatának akrofóniai indokolásával. Annál is inkább kívánatos, hogy megfelelő nyomra bukkanjunk, mert a lineáris B jeleivel való hasonlóság zsákutcába vezet. (Kipróbáltam!) Szerencsére a görög nyelv gazdag szókincse bőségesen ontja annak a szótőnek a különböző származékait, amely megfelelő megoldást kínál problémánkra. A TOREÓ ige jelentése „fúrni", ennek továbbképzett változata a TORNOÓ, TORNEUÓ 'kört rajzol, forgat, kerekít, esztergályoz'. A főnévi származékok közül említésre méltó a TORMOSZ 'lyuk, kerékagy', a TORUNÉ 'keverő, fakanál', és főleg a TORNOSZ 'körző, kör, kerekség' szó, amely végső soron a mi „turnus" szavunknak is az őse. E szócsalád távolabbi rokonságának egyes tagjai a latinon keresztül számos európai nyelvbe eljutottak: TORSIO, TORTURA 'csavarás' stb., de ebből származott (ófrancia közvetítéssel) az angol TURN ige is, vagy a késői utódok közül a spanyolból világ körüli turnéra indult TORNADO szó.

Bármit ábrázol is a korong „pajzs"-nak becézett jele, kétségtelen, hogy legfőbb jellegzetessége a kerekség. Akár pajzs volt ez a tárgy, akár más, bizonyára ez a tulajdonsága vált a forrásává elnevezésének is.

Vargha Dénes

A Mikrobazár rovatban a nem kereskedelmi célú egyéni hirdetések közlése ingyenes

A kereskedelmi célú apróhirdetések tarifája gépelt soronként (azaz 60 karakterenként) 300 forint. A terjedelem alapján így kiszámított összeget kérjük az Új Alaplap Kiadói Kft számlájára átutalni (OTP, 11701004–20171649), vagy postautalványon a kiadó címére elküldeni (1539 Budapest, Pf. 571), és feltüntetni, hogy „Új Alaplap, apróhirdetés". A befizetést igazoló szelvény másolatát — a hirdetési szöveggel együtt — a szerkesztőséghez (a kiadóéval azonos címre) küldjék el.

Szerzői jogokat sértő szoftverhirdetéseket nem közlünk le.

Bármilyen típusú szöveg fordítását vállalom angolról magyarra, magyarról angol nyelvre, illetve kiadványok látványtervezését, szerkesztését is. Cím: Lachner Zoltán, 1195 Budapest XIX., Jáhn Ferenc u. 14/a. Telefon: 157-0308.

OBJECTS 2.0 — objektumorientált programozás CLIPPER-ben. Tájékoztató kérhető az alábbi címen: Szűcs János, 4400 Nyíregyháza, Vasvári Pál u. 37. Tel.: (42) 437-331 vagy 465-666/1382-es mellék.

Adatmentés CD-re, streamerre; winchesterről, floppyról. Ugyanitt beszerzési tanácsadást, hálózattervezést és programkészítést is vállalok. Cím: Kovács Lajos, 1031 Budapest III., Vízimolnár u. 10. IV/33.

Stúdiómban megbízhatóan, ellenőrzötten lefordítom angol, német, francia és magyar nyelvről/nyelvre műszaki és közgazdasági folyóiratok cikkeit, hardver- és szoftverleírásait. Áfás számlát állítok ki. Cím: Szász György, 1035 Budapest III., Kórház u. 25. Tel.: 168-4874.

Alaplapcsere, memória-, winchester- és floppybővítés a helyszínen. MegaSoft. Telefon: 295-5085.

Négyszögjel generátor program PC-re ingyen! 1,44-es floppyt, válaszborítékot küldjön! Cím: Sipos András, 1118 Budapest XI., Radvány u. 10.

Használt laptopot vennék! Cím: Bródy Béla, 8000 Székesfehérvár, Eszperantó tér 3. I/3.

OPTI-495SLC (rev 1.1) alaplaphoz leírást keresek. Cím: Szőts Dávid, 1253 Bp., Pf. 30.

Keresek jogtiszta ChiWriter szövegszerkesztő programot. Cserébe megállapodás szerint küldök játék vagy egyéb programokat. Cím: Jakab Tamás, 4700 Mátészalka, Nagybányai u. 34.

Keresek átvételre számítógépes játékokat (csak floppyról). Vállalok számítástechnika tanítást alapfokon. Cím: Votisky András, 1021 Budapest II., Széher út 87. Telefon: (06-60)332-470.

Keresem más számítógépes magazinok mágneslemez mellékleteit (Chip, Computer Panoráma, Guru stb.) cserére, esetleg megvételre. Ugyanitt számítástechnikai oktatás lehetséges. Cím: Kovács Gábor, 3502 Miskolc II., Pf. 83.

486SX/40 MHz, 2 MB RAM, 1,2 + 1,44 FDD, 40 MB HDD, color SVGA monitor + szűrő, baby ház, Adlib hangkártya, 101 gombos billentyűzet, egér + alátét, joystick és újságok **eladók 75.000 Ft-ért.** Cím: Maczkó László, 8130 Enying, Rákóczi u. 4.

MONITOR SONICA 6690-es kapcsolási rajzát (Service Manual) keresem — kérem utánvéttel! Cím: Mizsei József, 6729 Szeged, Szabadkai út 106. Telefon: (06-30)587-011.

Keresek 40–100 MB-os winchestert, valamint 9 bites 1 MB-os RAM-okat. Telefon: 227-9417 15 óra után.

Schneider C.P.C. tulajdonosok! Kérlek, írjatok, ha bármilyen program, könyv, teszt vagy bármi olyan van a birtokotokban, ami a Schneider C.P.C. működését, használatát, előnyeit (hátrányait) mutatja be! Cím: ifj. Mester János, 6239 Császártöltés, Kiscsala 60.

Olcsón eladó egy 250 MB-os belső Conner streamer. Másfél éves, alig használt, jó állapotban van. Telefon: Fellner Ferenc 258-5778. BBS: (22.00-7.00 óráig) 258-5778.

INFORMÁCIÓKÉRÉS: 0711 ◄

„Van másik..."

Variációk egy témára

Tóth Dezső:

OS/2 Warp

Felhasználói ismeretek
ComputerBooks, 1995
394 oldal, 1680 Ft

Móricz Attila:

OS/2 Warp 3

Nyitott rendszerű képzés (távoktatás) oktatási segédlete
LSI Oktatóközpont, 1995
A magyar változatról:
326 oldal, 1620 Ft
Az angol változatról:
318 oldal, 1500 Ft

Czövek Gábor—Hofer Tamás András:

OS/2 Warp

Kézikönyv és referencia kezdőknek és haladóknak
IBM — IDG Books, 1995
352 oldal, árjelzés nélkül

A Warpról szóló könyvek ismertetése nem lenne teljes a történeti háttér megismerése nélkül. Az eseményeket magunk is átéltük, de talán nem mindig vettük észre, mi minden történt körülöttünk...

Öngól

A Windows 95 körüli nagy csinnadratta lecsengőben van. A hatalmas propagandakampánnyal annyit sikerült elérni, hogy sok felhasználó komolyan fontolóra veszi a 32 bites operációs rendszerre való áttérést. Néhány évvel ezelőtt ez még vágyálmaik között sem szerepelt, a hatalmas háttértárolók és az egyre nagyobb teljesítményű processzorok megjelenése azonban felvillantotta a lehetőséget.

Divatból persze nem szabad nagyobb változtatásba fejest ugrani (főleg ha azt sem tudjuk, van-e víz a medencében!). Egyedül a Win95 például nem elég indok egy generális hardver- és szoftvercserére. (Sokak szerint csak majd ha kijön annak 3. változata... arra már jobban lehet komoly fejlesztéseket is építeni.)

Két út van előttem...

A Win95 ígéreteivel mesterségesen felcsigázott — és részben kielégítetlenül maradt — várakozások arra késztetik az igényesebb felhasználókat, hogy alternatív megoldás után nézzenek. Igen tanulságosak azok a felmérések, amelyekre Ambrózy Gábor hívta fel a figyelmünket lapunk májusi számában: a Win95 megkérdezett felhasználóinak több mint a fele egy éven belül újabb váltásra készül. Mintegy kétharmad részük az OS/2-t részesíti előnyben, a többiek a Windows NT-t.

A Windows NT, el kell ismerni, beváltotta a hozzá fűzött reményeket, sokkal inkább, mint a sokkal nagyobb publicitást kapott fiatalabb testvér. Memóriaigénye ugyan saját kategóriájában is meghökkentően magas (minimum 16, de inkább 32 MB RAM, és gigabájtokban mérhető háttértár), de biztonságos adatkezelése, és más rendszerekkel való együttműködési készsége sokat nyom a latban.

Az igazi meglepetés nem is az NT viszonylagos előretörése volt, hanem a hír másik fele. Az, hogy a hovatovább már legyőzöttnek hitt vetélytárs, az OS/2 a Win95 újdonsült híveit is képes elhódítani.

A cserbenhagyott menyasszony

Az OS/2 fejlesztését még együtt kezdte el az IBM és a Microsoft 1987-ben, a DOS 3.3 megjelenésének évében. A kitűzött cél az volt, hogy közösen hozzák létre a DOS-t felváltani hivatott „második operációs rendszert" (ebből származott az Operating System/2, az OS/2 név). Az MS még némi hozományt is vitt a vállalkozásba — azoknak a visszhangtalanul maradt kezdeti próbálkozásoknak a tapasztalatait, amelyeket előzőleg szerzett egy ablaknyitogató grafikus felhasználói felület (bizonyos Windows) kifejlesztésében.

1990-ig ment is simán az OS/2 fejlesztése — évente kirukkoltak egy-egy újabb változattal. Elsőnek a 640 KB-os kaloda szorításából akartak kiszabadulni, ezért áttértek a védett módú futtatásra (1987). A következő évben bevezettek egy Windows-szerű grafikus kezelői felületet (1988), de már okulva a Microsoft tapasztalataiból. Majd a háttértár fájlkezelésének megújítása következett: sikerült minimálisra szorítani a fájlok töredezését, és gyakorlatilag kiküszöbölni a láncolási hibákat (1989). A 1990-es változat már annyira üzembiztos volt, hogy nagyvállalatok, bankok nyugodtan rábízhatták az új operációs rendszerre ügyvitelük gépesítését, ami DOS alatt még meglehetősen kockázatosnak látszott.

És ekkor következett be a szakítás. Abban egyetértett a két cég, hogy az objektumorientált programozás lehetőségeit kihasználva kell tovább haladni. Az IBM azonban többre becsülte az adatkezelés biztonságát, és elsősorban az „egy gép — egy program" kényszerétől való szabadulás irányában akarta folytatni a fejlesztést. A Microsoft egy tetszetősebb megoldás híve volt, és a gyors siker reményében választott öszvérmegoldásért inkább sutba dobta a közös koncepciót. Úgy döntött, hogy önálló operációs rendszer helyett megelégszik egy látszólag önálló, valójában azonban a DOS-ra épülő „pszeudo operációs rendszerrel", s ennek alapjául saját, régóta dédelgetett Windows-koncepcióját teszi meg.

Windowsos világ

A piac lélektanát, pillanatnyi befogadókészségét kétségtelenül a Microsoft mérte fel jobban. Harmadszori nekifutásra a Windows végre óriási sikert hozott Bill Gatesnek. Az OOP segítségével a fejlesztők ügyesen megoldották a rendszer bonyolult belső szervezését, bár az erőforrásokkal való gazdálkodás és az üzembiztosság egyáltalán nem nevezhető optimálisnak. Az új rendszer látványos megoldásai divatot teremtettek, és a DOS-t a felhasználók lassan már csak teherhordó szamárnak kezdték tekinteni.

A processzorok fejlődésének diadalmenete ekkor kezdődött igazán. A Windows eleinte sikeresen felszívta és csicsává konvertálta a hatalmasra duzzadt üres kapacitást, a 90-es évek közepére azonban már mindenki érezte egy igazi, önálló, és főleg biztonságos operációs rendszer hiányát. A Microsoft — immár a piac letarolása után — érthetően a Windows jegyében képzelte el a megújulást, de voltaképpen az elhagyott

útra próbált visszakanyarodni, minél kisebb presztízsveszteséggel.

Négy év — nagy idő...

És ez volt az a pont, ahol Bill Gates elszámította magát. Nem vette figyelembe, hogy az elmúlt négy év az IBM-nél sem múlt el tétlenül. Míg ő a Windows felszínének javítgatásával foglalkozott, addig az IBM kifejlesztette nagyságrendekkel üzembiztosabb saját operációs rendszerét, szabad lehetőséget teremtve DOS alatti és Windows alatti programok futtatása és fejlesztése számára is. Az egymástól élesen elválasztott partíciókban nem garázdálkodtak idegen programok, egy-egy program lerobbanása nem rántotta magával a mélybe az egész operációs rendszert. Eljött az idő, hogy még a Microsoft fejlesztői is inkább ebben a rendszerben dolgoztak, nem a sajátjukban.

1992 őszén az OS/2.1 már nemcsak a Win 3.1-es programokat futtatta tökéletesen, hanem egységes multimédia-felületével a hangnak, a filmnek, a játékprogramnak is a legmodernebb környezetet biztosította. 1993-ra változtattak azon a visszás helyzeten, hogy feleslegesen helyet kelljen foglalni, sőt jogdíjat kelljen fizetni a rendszerhez hozzáépített Windows kódokért — hiszen azok a legtöbb felhasználónál már megvoltak.

Színre lép a Warp

1994 végén azután, fejlesztési becenevét megőrizve, Warpként jelent meg az OS/2, igazi 32 bites, kistafírozott, könnyebben kezelhető, mégis kevesebb memóriát fogyasztó változatban, amelynek azóta már magyarra (és sok más nyelvre) honosított változata is van. És ami a legfontosabb, máris több-ezer alkalmazás készült rá, a Win95-re készült programoknak sokszorosa. Mint olvashattuk, még ebben az évben meglesz az újabb fejlesztés, a Merlin is, amelynek a Win95-re készített programok futtatása sem okoz problémát.

A fenti történeti háttér után lássuk magukat a könyveket. Célkitűzésük tulajdonképpen ugyanaz: teljes körű bevezetést adni a Warp világába. Érdekes viszont, hogy mennyire eltérő eszközökkel igyekeznek ezt a célt elérni.

Az alapos

Tóth Dezső könyve tűnik a legalaposabban kidolgozottnak. 80 oldalon keresztül a kezdő szintről indulva vezeti el olvasóit a Warp biztonságos kezelésének ismeretéig. Utána kétszer ekkora helyen tárgyalja egy-egy speciális terület problémáit, a Windows-alkalmazások működtetésétől a multimédia-alkalmazásokig. Bemutatja a háttértárak kezelésének, a nyomtatásnak, a rendszer telepítésének, finomhangolásának és üzembiztos működtetésének titkait. Ezután a könyv egyharmadát kitevő terjedelemben, gondos kidolgozásban részletesen ismerteti az OS/2 új és a DOS-ból átvett parancsait, szintaxissal, szemantikával, paraméterekkel, megjegyzésekkel és példákkal fűszerezve. A függelékekben az angol és a magyar változat használóinak nyújt hasznos segédeszközt: a két nyelv szerinti rendszerezésben külön-külön megadja a másik nyelvben használt terminust, és az oldalszámokat, ahol az adott parancs használata le van írva.



A kedélyes

Móricz Attila számítástechnikai tankönyvíróként mutatkozik be, s minden alkalmat felhasznál saját „Móricz-sorozatnak" nevezett könyvei propagálására. Kétségtelen erénye a szerzőnek, hogy írásban is meg tudja őrizni az élő nyelv élénkségét, s gondolatai cikázásával olvasóit is éberen tartja. Igyekszik kipróbálni a rendszer beállításának minden lehetőségét, s tapasztalatait az olvasóknak is átadni.

Bár a nyelvi lektor a legtöbb helyen gondos munkát végzett, bosszantóan sok helyesírási hiba is maradt a szövegben. Az elhagyott vesszők (az amit; nézzük meg milyen; tudjuk hová; láttuk hogyan; stb.), kompenzálásként a feleslegesen kitett sem ritka. Különösen a III. részben sok a kusza, fésületlen fogalmazás. Helyenként az idegen szavakkal is baj van (pl. „analógiája" helyett „szinonimája" szerepel, 259. old.).

És talán azt is érdemes lenne megnézni, hogy egy felsőoktatási tankönyv bevezetésének szabályozva vannak-e a kritériumai.

A bennfentes

Csak az összehasonlítás teljessége kedvéért említem meg Czövek Gábor és Hofer Tamás András könyvét, mivel könyvárusi forgalomba nem került. Ez is gondos munka eredménye, különösen referenciaként, esetenkénti fellapozásra kitűnő. Folyamatos olvasásra már sokkal kevésbé. Tájékozódni könnyű benne (jó tárgymutatójának, kereszthivatkozásainak és gondos tipográfiájának köszönhetően), és elszórva sok olyan információmorzsát tartalmaz, amit másutt nem találtam meg. Látszik, hogy a szerzők gazdagon éltek azokkal a háttér-információkkal is, amelyeket közvetlenül a cégtől kaptak, és feltehetően maguk is alaposan tesztelték a programrendszert. Több helyen felhívják például az olvasó figyelmét arra, ha egyik vagy másik funkció nem működik (vagy nem úgy működik) a magyar változatban, amint az elvárható volna.

Egyébként ez az a könyv a három közül, amely lemezmellékletet is tartalmaz (mindjárt kettőt is), gondosan megtervezett tartalommal. Található rajtuk kibővített parancsértelmező, helytakarékos munkafelület-kezelő, egy Norton Commander-szerű és egy még szélesebb körű szolgáltatást nyújtó fájlkezelő, OS/2-n futtatható McAfee-féle víruskereső és egyéb nyalánkságok.



Nem lehetne ezt a könyvet a nagyközönség számára is hozzáférhetővé tenni?

Vargha Dénes

Windows alól, Windows alá

A Clipper új ruhája

A Clipperben fejlesztők széles tábora meglehetős bizonytalanságban volt a jövőjét illetően, amikor kedvenc fejlesztőrendszerét annak régi gazdájától megvásárolta a Computer Associates (CA). Egy ideig mintha megszakadt volna az addig folyamatos fejlesztés, lassanként kezdett csak csordogálni a CA-Clipper márkanévvel propagált, újonnan feltáruló forrás.

Az első nagyobb probléma, amelyet a CA megoldott, a 640 Kbájt feletti memória használata volt. Az 5.2 változathoz már elérhető volt az ExoSpace memóriakezelő, majd az 5.3-mal elkezdődtek a DOS-tól való elszakadási hadműveletek. Itt már Windows alatt egerészve lehetett folytatni a fejlesztést, de a tárgyprogram még DOS alá készült.

Az óvatosság tökéletesen érthető: köztudomású, hogy a Windows alatti programfuttatás hihetetlen mértékben le tudja lassítani a futást, ami katasztrofális következményekkel járhat komolyabb adatbáziskezelést kívánó feladatoknál.

Az új út keresése vezetett el egy gyökeresen eltérő rendszer megalkotásának gondolatához. Az új rendszer a keresztségben a Visual Object nevet kapta, s fennen hirdetett célja egy hatékony és könnyen kezelhető, „Windows alól, Windows alá" típusú programgenerátor kialakítása lett.

Az isteni szikra tulajdonképpen más fejéből pattant ki, mert a Visual Basic példája adta az ihletet, de ennek kezdeti hibáin okulva készítette el a CA fejlesztőgárdája első vizuális fejlesztőeszközét, a CA-Realizert. A fejlesztők rengeteg ügyes újításával, hatalmasra felduzzasztott makrókönyvtárával ez a rendszer jobb lett, mint példaképe, de igazán elterjednie mégsem sikerült. Végtére is fejlesztői kénytelenek voltak belátni, hogy a Basic nyelv nem a legjobb alap a komolyabb rendszerek kifejlesztéséhez.

A Visual Objects sokkal inkább az objektumorientált C nyelv lehetőségeit próbálja a háttérben kihasználni. Eközben számtalan kényelmi és biztonsági eszközzel segíti a vizuális programfejlesztést, s nem feledkezik meg arról sem, hogy a hagyományos Xbase adatkezelésen kívül az SQL táblák objektumorientált kezelői felületét is hozzáférhetővé tegye. A fejlesztés legfőbb szempontja — az adatbáziskezelés tetemes gépidő-igényességére való tekintettel — a hatékonyság fokozása, de legalább a DOS alatti szint megőrzése volt. Közben persze a clipperes fejlesztői tábor kialakult szokásait, beidegződéseit is tekintetbe kellett venni, fokozatosan vezetve át őket egy új stílusú programozásba.



Az új programtermék sok kellemes tulajdonsága ellenére még további fejlesztésre szorul: ez abból is látszik, hogy az 1.0 variánst gyors egymásutánban követte az 1.0a, 1.0b és 1.0c. Ezek mindegyike elég lényeges módosításokat tartalmazott, bár az indoklás legtöbbször nem a hibák kijavítása, hanem a hatékonyság fokozása volt. Jó tudni, hogy a felhasználók egyre növekvő tábora állandó információcsere-kapcsolatban van a CompuServe hálózaton keresztül (GO VOFORUM). Ez különösen azok számára fontos, akik Windows 95 vagy Windows NT alatt futtatják a CA-VO-t, mivel a Debugger az új környezetben csak korlátozott mértékben használható. (A hosszú neveket sem hajlandó a rendszer kiszolgálni, ennek magyarázata azonban másutt, a Microsoft Resource Compilerében keresendő.)

Dr. Dedinszky Ferenc

CA-Visual Objects

Objektumorientált XBASE programnyelv Windows alá
ComputerBooks, 1995
272 oldal, 1559 Ft
(+ lemezmelléklet 400 Ft)

Dedinszky Ferenc könyvében kezdőknek is érthető módon ecseteli a CA-VO előnyeit, magyarázza használatát, nem feledkezve el a profi felhasználók szakszerű tájékoztatásáról sem. Ezen a könyvén azonban látszik, hogy elmaradtak az utolsó simítások. Terminológiája és a könyv struktúrája menet közben alakult ki, így helyenként például nem létező vagy más néven szereplő fejezetekre utal, kitöltetlenül maradnak a kipontozott helyek, angolul hivatkozik előzőleg csak magyarul megnevezett ikonokra stb. Sőt az is előfordul, hogy hibásan adja meg egyes események magyar fordítását (47. old.).

Kellemetlenebb következménye a befejezetlenségnek, hogy használhatatlanná vált a még félkész anyagból automatikusan készített tárgymutató. Az oldalszámok csak véletlenszerűen stimmelnek, s például külön tárgyszóként vette fel a tárgymutató-készítő program az elírásból keletkezett variánsokat is. (Böngészdék mellett Bömgészdék, Kódblokk mellett Lódblokk stb.) Ráadásul a tárgyszókészítésre kijelölt szavak, kifejezések köre is esetleges és következetlen, ami írás közben persze érthető, utólag azonban kijavítható lett volna.

Sokat segít a rendszer megismerésében a könyv bőséges illusztrációs anyaga. Sajnos, az illusztrációs lemezmellékletet csak külön lehet megvásárolni a szoftver disztribútorától, a PC Szoftvertől, ahol azonban — tapasztalatom szerint — a segítőkész szakértők gyakorlati kérdésekben is készségesen állnak a felhasználók rendelkezésére.

Vargha Dénes

Mü govorim po-internetszki

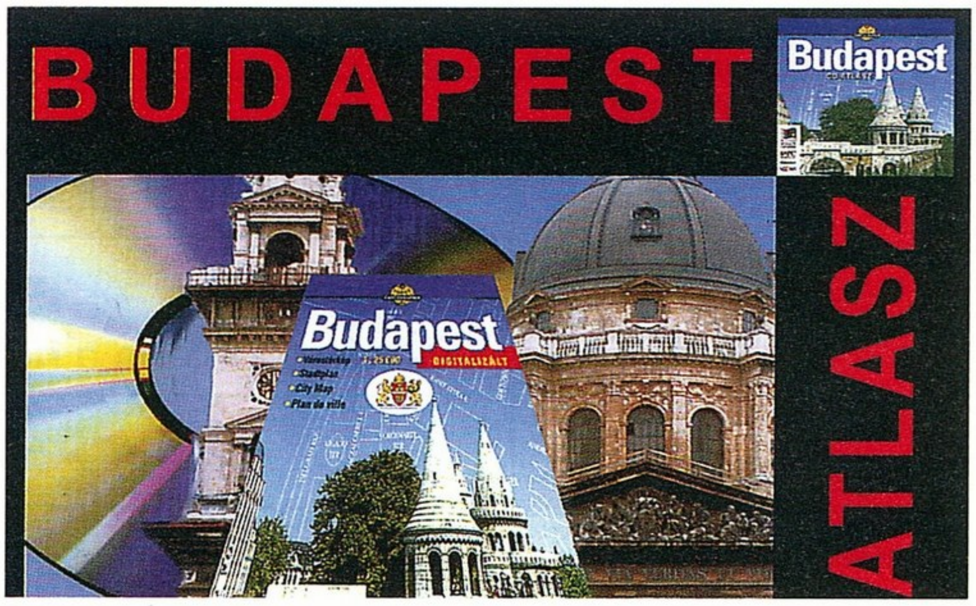
Lehet, hogy néhány Internet-búvár „49ers"- vagy „Giants"-drukker feloldhatatlan dilemma elé kerül: magán a meccsen rágja-e tövig a körmét, vagy helyette a stadionba kihelyezett, ingyen használható touch screen terminálon keresztül bóklásszon az Internet rejtett bugyraiban?

A „Beszélünk Internetül" kioszkok rendszere — amelyet a 3Com egy sportstadionnak nyújtott támogatás keretében üzemeltet — a Fehér Ház vagy a párizsi Louvre honlapján túl persze a különböző sportegyesületek honlapját is elérhetővé teszi. (Bigott sportrajongók, figyelem!)

Rövidesen tehát teljesen természetes dologgá válhat, hogy a másik stadionban játszó riválisok eredményét — vagy akár magát a játékot is — online követhesse bárki, s az ottaniak ismeretében biztassa a saját csapatát. A fentiek fényében méltán keresztelték át tavaly ősszel 3Com Parkra a San Francisco Candlestick kerületében álló stadiont.



Budapest CD-Atlasz



Ha a korszerű hardver- és szoftvertechnikai lehetőségek messzemenő kiaknázásának lehetőségét tekintjük, alighanem csillagos ötöst érdemel a Budapest CD-Atlasz megvalósítása. Ha viszont a napi praktikumot tekintjük, még jó ideig eltart, mire az így együvé gyűjtött információk „aprópénzre válthatók" lesznek. Merthogy mi is ez a Budapest CD-Atlasz?

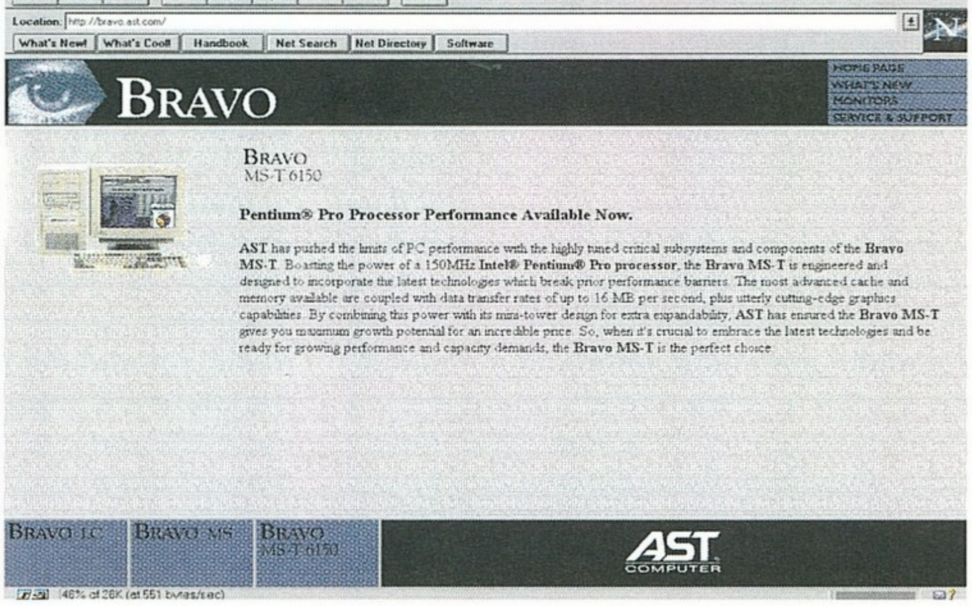
Digitalizált, azaz vektorgrafikus számítógépes térkép, multimédiás térinformatika. Arra való (pláne az ajánlott hardver ismeretében: Pentium, 16 Mbájt RAM, Win95 vagy NT, true color kártya+monitor, 4-szeres CD-ROM), hogy az ember otthon vagy munkahelyén kényelmesen feltérképezhesse azt az útvonalat, amelyre kíváncsi. Arra viszont korántsem való, hogy útközbeni eligazítást nyújtson. Még nem.

A CD szolgáltatásai (egyszerű és gyors zoom, frissíthető adatok, teljes utca- és tömegközlekedési hálózat, nevezetességek fotógyűjteménnyel és videóval, felhasználó által beilleszthető objektumok stb.) viszont már most előrevetítenek olyan lehetőségeket, amelyek a későbbi mobil felhasználás területén lesznek igazán kiaknázhatók.

A rendszer igazi alkalmazási diadalát alighanem abban a jövendő korszakban éri meg, amikor a térképi információk a szoftveres-hardveres kapukon keresztül teljesen természetesen funkcionálnak majd más, akár nagyobb, akár kisebb (tér)informatikai rendszerek elemeként. (Ilyen lehet, mondjuk, egy autós szakértői rendszer, amely „robotpilótaként" a legrövidebb útvonalon juttatja el utasát a nagykörúti tankcsapdákon át a Tűzoltó utcából a Műegyetemre.)

Bravo, Ascentia!

Nagy rendszerteljesítményt igénylő alkalmazások számára kínálja az AST legújabb Bravo modelljét. A 150 MHz-es Pentium Próval szerelt Bravo MS-T 6150-es minitorony maximális bővíthetőséget kínál, hét drive-csatlakozással, négy 32 bites PCI bővítőhellyel, és három 16 bites ISA



bővítőhellyel. További kulcsfontosságú funkciók: alaplapba integrált 1,6 Gbájtos IDE hard drive, 16 Mbájt (128 Mbájtig bővíthető) memória, 256 Kbájt cache, PCI 64 bites ATI grafikus vezérlő, 2 Mbájt VRAM (4-re bővíthető), 16 bites SoundBlaster Vibra 16 audiorendszer.

Az új Bravóval egy időben két új notebook-család is megjelenik a magyar piacon. Az Ascentia J (100 MHz-es Pentiummal) tervezésénél mindenekelőtt az ergonómiai szempontok érvényesültek: a könnyű kezelhetőséget szolgálja a SmartPoint trackstick és a speciális Windows'95 gomb is. Az Ascentia J notebook integrált 16 bites hangkártyával, valamint 800X600-as DualScan STN, illetve TFT aktív mátrix kijelzővel kerül forgalomba.

A P50 felségjelű AST notebookot elsősorban a 133 MHz-es Pentium, valamint a sztereo SoundBlaster, illetve a 4-szeres sebességű CD-ROM-drive különbözteti meg a „kistestvér" J családtól. A forgalomba kerülő új gépek árairól nincs információnk.

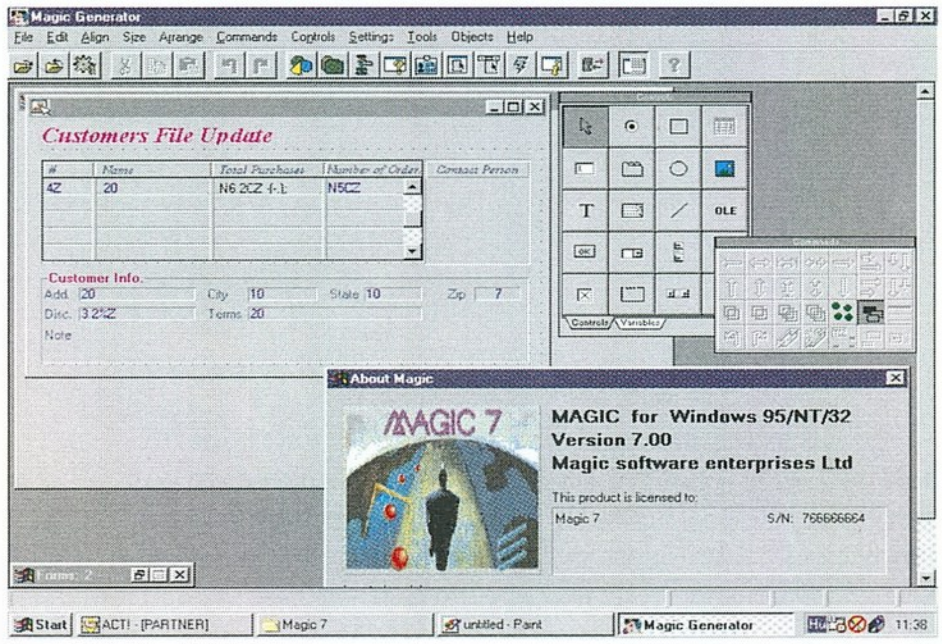
Magic-hét

Már megszoktuk, hogy évente ilyentájt lát napvilágot a hír, miszerint az alkalmazásfejlesztő eszközök soros bajnokságán már megint a Magic vitte el a pálmát. Azok kedvéért, akik nem ismerik ezt a rendszert, röviden csak annyit, hogy a Magic egy nagyon hatékony, RAD (Rapid Application Development = gyors alkalmazásfejlesztő) eszköz, amely hordozható, sőt bonyolult kliens-szerver alkalmazások fejlesztését is lehetővé teszi. Egyedi módon közelíti meg az alkalmazásfejlesztést, mivel magát az alkalmazást is egy adatbázis igen absztrakt elemeként kezeli. Ez az adatbázis adatként tárolja az alkalmazás összes jellemzőjét, definícióját, és az alkalmazás egyedi értékeit — a képernyőtervektől a működési szabályokig és logikáig. Futtatáskor a Magic „motorja" értelmezi ezt a különleges adatbázisfájlt.

Figure 1. Spiral Development Model



A Magic és a hozzá hasonló RAD eszközök megjelenése a fejlesztési technológiák kidolgozóit is új „filozófiák" keresésére sarkallta. A RAD metodika legismertebb alternatívája a „vízesés modell", ahol a fejlesztés lineárisan és teljes szélességben halad — pontosan definiált fázisokon át —, és mint egy folyó tör előre a zuhatagokon keresztül. Viszonylag újabb modell a „spirális fejlesztés", amely ugyancsak a kockázatcsökkentés alapvető igényéből indul ki, de magába integrálja a RAD eszközök lehetőségeit. E modellben az alábbi tevékenységek ismétlődnek ciklikusan, egyre szélesebb részt lefedve a megoldandó problémából: célok megfogalmazása, kockázatelemzés, prototípus készítése, tesztelhető modell elkészítése, tervezés és fejlesztés, a következő ciklus tervezése.



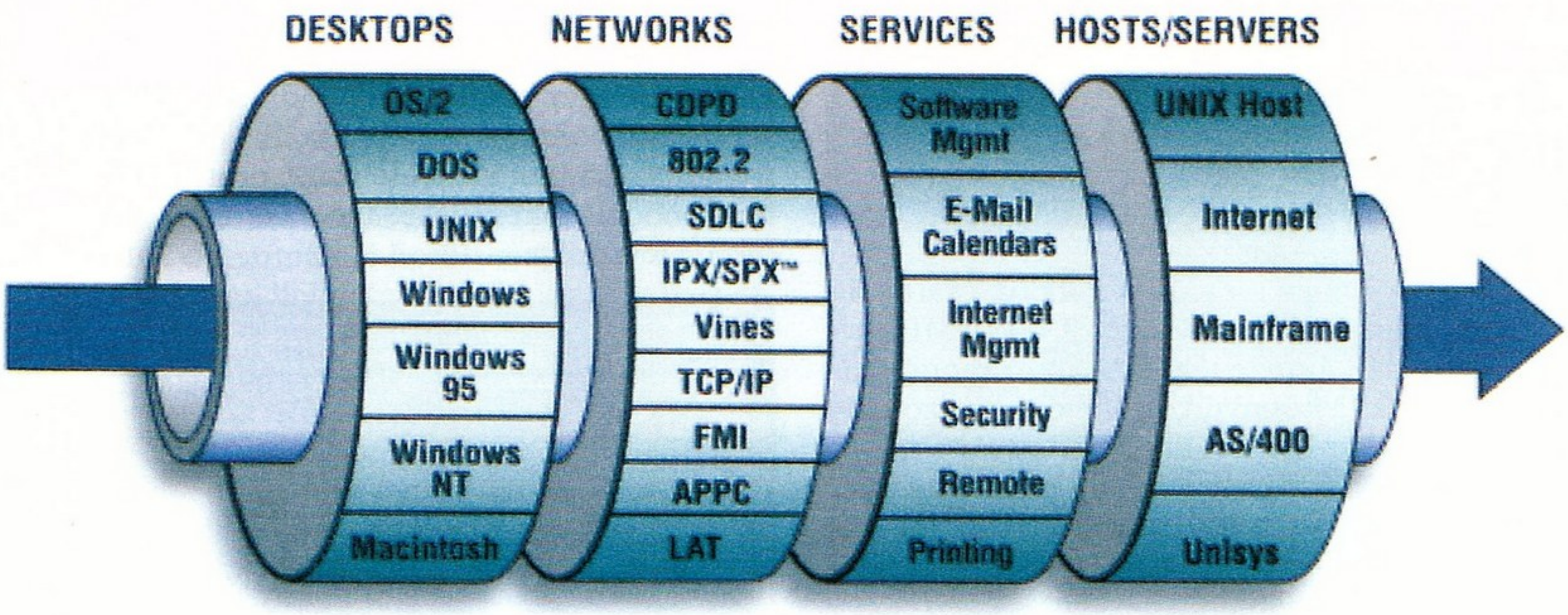
Érdemes szólni a Magic-filozófia egyik fő előnyéről: a fejlesztés leginkább azért gyors, mert szinte egyáltalán nem kell kódot írni. A fejlesztő az alkalmazást táblázatok kitöltésével definiálja, párbeszédablakokon és választásokat kínáló szótárakon keresztül — mindössze 13 utasítás segítségével.

Nos, lehet, hogy a mostani bejelentés nem hat majd olyan bombasztikusnak, mint a korábbiak. Neve dacára csodát, varázslatot nem is kell várni attól, hogy egyéves fejlesztőmunka eredményeként megszületett a fejlesztőeszköz új, 7-es verziója, de a „csodáknál" általában többet ér egy megalapozott, logikus továbbfejlesztés. A Magicben fejlesztők számára mindenképpen megnyugvást jelenthet, hogy az általuk választott rendszer is tartja azt a tempót, amelyet a web-robbanás kényszerít ki az alkalmazásfejlesztő eszközök fejlesztőiből.

Szinte alapkövetelménynek számít ma már az Internet-kapu megvalósítása (benne is van az új verzióban a web-link), a Magic-fejlesztők nem is erre helyezték a hangsúlyt, az Internet-kapcsolat inkább csak valami szükségszerű „mellékes". Az új változat legfőbb jellegzetességeként az elemzők inkább a komponensek újbóli felhasználhatóságának, valamint az OLE 2.0 támogatásának minden eddigit meghaladó mértékét, illetve a grafikus felületet és a kliens/szerver architektúra kezelését emelik ki. Az új, 32 bites fejlesztőeszköz a Windows 95, illetve a Windows NT platformra van kihegyezve, de a tervek szerint rövidesen elkészülnek más rendszerek új változatai is.

A szürke eminenciás

Ha olyan kérdéssel „vizsgáztatnánk" a magyar számítástechnikusokat, hogy sorolják fel a világ 10 legnagyobb szoftveres cégét, feltehetően nem esne nehezükre összehozni ilyen listát, de telitalálatot elég kevesen érnének el, a túlnyomó többség pedig egészen biztosan ki is hagyná a 7. helyezettet. E hetedik egy (nálunk még) szürke eminenciás, az Attachemate. Most, hogy a magyar piacon is aktivizálódik (disztribútora az ITD Kft), ismertebbé fog válni, és amit kínál, az egyre inkább felértékelődik.



Az Attachemate „emulációs cégként" vált ismertté, amelynek termékei lehetővé teszik, hogy adataink végig zöld jelzést kapjanak, akkor is, ha a bejárandó útvonalon az operációs rendszerek, a hálózati szabványok, a szolgáltatási formák és a kiszolgálógépek teljesen eltérőek. (Lásd a fenti ábrát.) Ráadásul a közelmúltban igen jó szakmai erősítésre tettek szert a „The Wollongong Group" megvásárlásával. Akiknek ez utóbbi sem cseng ismerősen, azoknak csak annyit, hogy a Wollongong nevéhez fűződik a TCP/IP szabvány kifejlesztése, újabb sikertermékük pedig a nagy teljesítményű Internet-hozzáférést nyújtó Emissary.

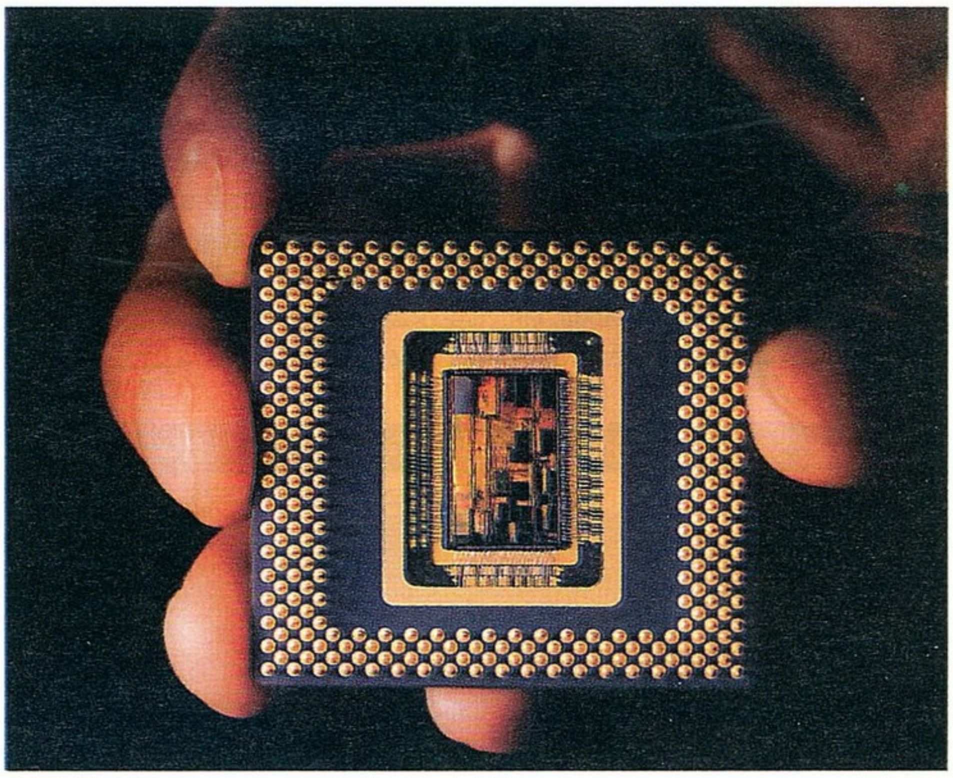
Az Attachemate fejlesztési irányát jól mutatja, hogy emblémájába is felvette „az intranet vállalat" állandó jelzőt. „Extra!" termékcsaládja a vállalatoknál eddig is igen széles skálán és nagyon heterogén számítástechnikai környezetben segítette elő az alkalmazások és az adatok megbízható hozzáférését, az internetes technológia „intranetesedése", vagyis a vállalati belső hálózatok működésére való adaptálása pedig óriásira tágítja az Attachmate programjainak piacát, különösen ott, ahol a vegyes rendszerek dominálnak. A hálózatok összekapcsolásának szoftveres megoldásaira specializálódott Attachemate a jövő nagy lehetőségének a keresőprogramokat (a browsereket) tekinti, és „erre a lapra tesz".

A Pentium alternatívája

A személyi számítógépek világában az Intel determináló szerepe a kisebb proesszorgyártókat arra szorítja, hogy alkalmazkodjanak a piacvezető által kialakított normákhoz. Ilyen lényeges szempont most például a Windows operációs rendszerek és grafikus felületek zavartalan futtatása, és a Pentiummal való csereszabatosság. Ennek jegyében hozta ki a negyedik legnagyobb amerikai integráltáramkör-gyártó cég, az AMD (Advanced Micro Devices) az AMD5$_K$86 ötödik generációs processzorcsaládot, amely a Windows-változatok mellett kompatibilis az x86-os operációs rendszerek és szoftverek teljes skálájával, a Novell NetWare hálózati rendszerrel, a Unix-szal, az OS/2 Warppal, stb.

Az AMD5$_K$86 az első saját fejlesztésű mikroprocesszor az AMD termékei közül. Kitűnő teljesítménye annak köszönhető, hogy az x86-os utasításokat RISC utasításokra fordítja le (ROP), és szuperskalár felépítésű egységei egy órajelre akár 6 ROP utasítást is végre tudnak hajtani, illetve mindig négyet vesznek újra elő.

A félmikronos technológiával készülő új processzor kettős típusmegnevezése (dual marking) számunkra kissé körülményesnek tűnik. Az egyikben szerepel egy új teljesítménymutató, a P-arány (P = performance = teljesítmény), ami azt jelenti, hogy például a 100 MHz-es AMD5$_K$86-P133 processzor teljesítménye megfelel egy 133 MHz-es Pentiuménak. De ez a típusnév nem árulja el az AMD-processzor tényleges frekvenciáját. Arra egy másik, a fejlesztés idején használt (és szerintünk logikus) elnevezés szolgál: AMD-K5-100. Félreértések elkerülésére, ez a kettő ugyanazt a processzort jelenti! Lehet, hogy az AMD menedzserei szerint ez a sajátos névadási szisztéma kiváló marketingeszköznek bizonyul majd, de az AMD magyarországi sajtótájékoztató-



jának közönsége előtt nem aratott átütő sikert. A dolgok persze nem itt dőlnek el, és az a fontos, hogy főképpen olyanok igazodjanak el a kódrejtvényekben, akik a processzorokat beépítik a gépekbe. Nekünk elég az is, hogy azonos teljesítményű számítógéphez jóval alacsonyabb áron tudunk hozzájutni.

Az AMD nemcsak a csúcs közelében, hanem lejjebb is figyelembe veszi a piac „többrétegűségét". Például nem állítja le a jól bevált 486-os processzorok gyártását sem, hiszen bizonyos célokra az tökéletesen elegendő (még a gazdag országokban is), másrészt a Föld kevésbé fejlett régióiban a vásárlóerő a jelek szerint hosszabb távon sem éri el azt a szintet, hogy csak a pentiumos gépek iránt legyen kereslet.

Az 1969-ben alapított AMD központja a kaliforniai Sunnyvale, de gyárai vannak Texasban, Angliában, Japánban, Szingapúrban, Thaiföldön és Malajziában is.

Intergraph *InterServe* WEB-szerverek



Az első teljes WEB-szervercsalád, mely a Microsoft Windows NT szerverén alapszik, és egyesíti a PC és UNIX szerverek legjobb tulajdonságait, most teljes megoldást ajánl a vállalatok WEB-igényeinek kielégítésére.

Powered by Intergraph InterServe

Intergraph *InterServe* WEB-szerverek

Típus	InterServe WEB-30	InterServe WEB-300	InterServe Firewall-300	InterServe Web-610	InterServe Web-630
Processzor	133 Mhz Pentium	150 Mhz Pentium Pro	150 Mhz Pentium Pro	150 Mhz Pentium Pro	Kettő 200 Mhz Pentium Pro
Upgrade	Helyszínen kettő proc.-ra	Helyszínen kettő proc.-ra	Helyszínen kettő proc.-ra	Helyszínen kettő proc.-ra	Helyszínen négy proc.-ra
Ház	Asztali	Asztali	Asztali	Torony	Torony
RAM	32 MB	32 MB, ECC	32 MB	64 MB, EEC	64 MB, EEC
RAM-bővíthetőség	256 MB-ig	256 MB-ig	256 MB-ig	1 GB-ig	1 GB-ig
Hard diszk, Fast SCSI-2	1 GB	2 GB	2 GB	4 GB	4 GB
Harddiszk-bővíthetőség	20 GB-ig	20 GB-ig	20 GB-ig	700 GB ig	700 GB-ig
Ethernet	10 Base-T	10 Base-T	10 Base-T és 100 Base-T	100 Base-T	100 Base-T
Grafikus gyorsítókártya	G95, 2 MB RAM, bővíthető 8 MB-ig	G95, 2 MB RAM	G95, 2 MB RAM	G95, 2 MB RAM, bővíthető 8 MB-ig	G95, 2 MB RAM bővíthető 8 MB-ig
Alkalmazás	Transcend	Transcend	Firewall	Transcend	Transcend
Alapszoftver	Microsoft Windows NT Server 3.51, IIS, SMT, POP, DNS, HTML szerzői eszközök				

Az InterServe WEB-szervercsaládból kiválaszthatja azt a megfelelő konfigurációt, amelynek segítségével biztosan építhet az Internet kapcsolataira.
Hívjon vagy keressen fel bennünket!

INTERGRAPH **Magyarország Kft.** 1149 Bosnyák tér 5. ☎ (36-1) 163-3888

INFORMÁCIÓKÉRÉS: 0713 ▲